

日本国特許庁
JAPAN PATENT OFFICE



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application:

2001年 3月19日

RECEIVED

JUL 09 2002

出願番号
Application Number:

特願2001-079353

Technology Center 2100

ST.10/C]:

[JP2001-079353]

出願人
Applicant(s):

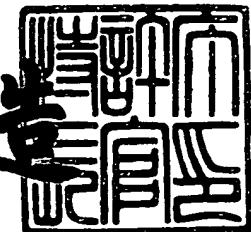
株式会社東芝

CERTIFIED COPY OF
PRIORITY DOCUMENT

2002年 3月29日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



【書類名】 特許願

【整理番号】 A000100934

【提出日】 平成13年 3月19日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/00

【発明の名称】 コンピュータシステムによる情報処理サービスの提供方法、プログラム、情報処理サービス提供システム

【請求項の数】 17

【発明者】

【住所又は居所】 東京都府中市東芝町1番地 株式会社東芝府中事業所内

【氏名】 大森 麻理

【発明者】

【住所又は居所】 東京都府中市東芝町1番地 株式会社東芝府中事業所内

【氏名】 松本 茂

【発明者】

【住所又は居所】 東京都府中市東芝町1番地 株式会社東芝府中事業所内

【氏名】 守安 隆

【発明者】

【住所又は居所】 東京都府中市東芝町1番地 株式会社東芝府中事業所内

【氏名】 佐波 公夫

【特許出願人】

【識別番号】 000003078

【氏名又は名称】 株式会社 東芝

【代理人】

【識別番号】 100058479

【弁理士】

【氏名又は名称】 鈴江 武彦

【電話番号】 03-3502-3181

【選任した代理人】

【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

【選任した代理人】

【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1
【物件名】 要約書 1
【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 コンピュータシステムによる情報処理サービスの提供方法
、プログラム、情報処理サービス提供システム

【特許請求の範囲】

【請求項1】 コンピュータシステムによる情報処理サービスの提供方法であって、

、ユーザにネットワークを経由して提供されるサービスに対して付加価値サービスを提供するためのイベントを検出するステップと、

前記イベントが検出された場合に、前記付加価値サービスを提供するデータ処理手段を呼び出すために用意されたライブラリを参照するステップと、

前記ライブラリの参照結果を利用して前記付加価値サービスを提供するデータ処理手段に処理実行を依頼するステップと

からなる情報処理サービスの提供方法。

【請求項2】 コンピュータシステムによる情報処理サービスの提供方法であって、

ユーザにネットワークを経由して提供されるサービスの運用状況に関するデータを記憶手段に記憶するステップと、

そのサービスに対して付加価値サービスを提供するデータ処理手段に対して、前記記憶手段に記憶された運用状況に関するデータを出力するステップとからなる情報処理サービスの提供方法。

【請求項3】 コンピュータシステムによる情報処理サービスの提供方法であって、

ユーザにネットワークを経由して提供されるサービスへのリクエストを取得するステップと、

取得されたリクエストに応じて、そのサービスに対して付加価値サービスを提供するデータ処理手段に、処理実行を依頼するステップとからなる情報処理サービスの提供方法。

【請求項4】 コンピュータシステムによる情報処理サービスの提供方法であって、

ユーザにネットワークを経由して提供されるサービスの処理に基づいて発生する付加価値サービスへのリクエストを取得するステップと、

取得されたリクエストに応じて、当初のサービスに対して付加価値サービスを提供するデータ処理手段に、処理実行を依頼するステップと
からなる情報処理サービスの提供方法。

【請求項5】 請求項3又は請求項4記載の情報処理サービスの提供方法において、

リクエストの内容と処理実行依頼先とを対応付けたテーブルを参照し、取得されたリクエストの内容に対応するデータ処理手段に処理実行を依頼することを特徴とする情報処理サービスの提供方法。

【請求項6】 請求項1乃至請求項5のいずれかに記載の情報処理サービスの提供方法において、

前記データ処理手段は、複数のサービスに共同で利用されることを特徴とする情報処理サービスの提供方法。

【請求項7】 請求項1乃至請求項6のいずれかに記載の情報処理サービスの提供方法において、

複数のデータ処理手段が連携して動作することでサービスに対して付加価値サービスを提供することを特徴とする情報処理サービスの提供方法。

【請求項8】 請求項1乃至請求項7のいずれかに記載の情報処理サービスの提供方法において、

ユーザに提供されるサービス毎に、前記データ処理手段の使用状況に応じた費用を費用計算手段によって計算するステップを附加したことを特徴とする情報処理サービスの提供方法。

【請求項9】 コンピュータに、

ユーザにネットワークを経由して提供されるサービスに対して付加価値サービスを提供するためのイベントを検出する機能、

前記イベントが検出された場合に、前記付加価値サービスを提供するデータ処理手段を呼び出すために用意されたライブラリを参照する機能、

前記ライブラリの参照結果を利用して前記付加価値サービスを提供するデータ

処理手段に処理実行を依頼する機能

を実現させるためのプログラム。

【請求項10】 コンピュータに、

ユーザにネットワークを経由して提供されるサービスの運用状況に関するデータを記憶手段に記憶する機能、

そのサービスに対して付加価値サービスを提供するデータ処理手段に対して、記憶された運用状況に関するデータを出力する機能
を実現させるためのプログラム。

【請求項11】 コンピュータに、

ユーザにネットワークを経由して提供されるサービスへのリクエストを取得する機能、

取得されたリクエストに応じて、そのサービスに対して付加価値サービスを提供するデータ処理手段に、処理実行を依頼する機能
を実現させるためのプログラム。

【請求項12】 コンピュータに、

ユーザにネットワークを経由して提供されるサービスの処理に基づいて発生する付加価値サービスのリクエストを取得する機能、

取得されたリクエストに応じて、当初のサービスに対して付加価値サービスを提供するデータ処理手段に、処理実行を依頼する機能
を実現させるためのプログラム。

【請求項13】 請求項11又は請求項12記載のプログラムにおいて、リクエストの内容と処理実行依頼先とを対応付けたテーブルを参照し、取得されたリクエストの内容に対応するデータ処理手段に処理実行を依頼することを特徴とするプログラム。

【請求項14】 情報処理サービスを提供するシステムであって、ユーザにネットワークを経由して提供されるサービスに対して付加価値サービスを提供するためのイベントを検出するイベント検出手段と、前記イベント検出手段によってイベントが検出された場合に、前記付加価値サービスを提供するデータ処理手段を呼び出すために用意されたライブラリを参照

する参照手段と、

前記参照手段によるライブラリの参照結果を利用して前記付加価値サービスを提供するデータ処理手段に処理実行を依頼する依頼手段とを具備したことを特徴とする情報処理サービス提供システム。

【請求項15】 情報処理サービスを提供するシステムであって、

ユーザにネットワークを経由して提供されるサービスの運用状況に関するデータを記憶する記憶手段と、

そのサービスに対して付加価値サービスを提供するデータ処理手段に対して、前記記憶手段によって記憶された運用状況に関するデータを出力する出力手段とを具備したことを特徴とする情報処理サービス提供システム。

【請求項16】 情報処理サービスを提供するシステムであって、

ユーザにネットワークを経由して提供されるサービスへのリクエストを取得するリクエスト取得手段と、

前記リクエスト取得手段によって取得されたリクエストに応じて、そのサービスに対して付加価値サービスを提供するデータ処理手段に、処理実行を依頼する依頼手段と

を具備したことを特徴とする情報処理サービス提供システム。

【請求項17】 情報処理サービスを提供するシステムであって、

ユーザにネットワークを経由して提供されるサービスへの処理に基づいて発生する付加価値サービスのリクエストを取得するリクエスト取得手段と、

前記リクエスト取得手段によって取得されたリクエストに応じて、当初のサービスに対して付加価値サービスを提供するデータ処理手段に、処理実行を依頼する依頼手段と

を具備したことを特徴とする情報処理サービス提供システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、サービス提供者を支援するコンピュータシステムによる情報処理サービスの提供方法、プログラム、情報処理サービス提供システムに関する。

【0002】

【従来の技術】

サービス提供者とは、例えばビジネスに利用される業務ソフトウェアなどのアプリケーション・ソフトウェア（以下、アプリケーションという）の機能あるいはそのアプリケーションによる出力結果をユーザ（顧客や他サービス）に提供する事業者あるいはその機能である。

【0003】

ユーザは、そのサービスを利用する主体であり、人間あるいは他のシステムである。

【0004】

例えば、A S Pのサービスとそのユーザで考える。A S Pでは、ユーザは、自己のパソコンにアプリケーションをインストールしなくとも、サービス提供者のレンタルアプリケーションを利用することでそのアプリケーションを利用した作業を実施可能である。これにより、ユーザ側では、企業の情報システム部門の大きな負担となっていたインストールや管理、アップグレードにかかる費用・手間を節減することができる。

【0005】

従来のA S Pでは、統合業務パッケージなどの大規模な業務システムがレンタルの対象であったが、近年ではワープロや表計算などの日常頻繁に使われるアプリケーションもレンタルされるようになりつつある。

【0006】

なお、実際には、ユーザのアクセス方法としてW e b ブラウザを利用しない場合もある。W e b ブラウザを利用しない場合の例としては、携帯電話専用ブラウザを利用する方法がある。また、期限付きのソフトウェアをダウンロードしてユーザ側のハードディスクなどに保存し、ユーザがこのソフトウェアを利用する方法もある。

【0007】

サービス提供者には、サーバのダウンや異なるユーザ間での情報漏洩が許されない。そのため、サービス提供者はサーバの設置場所としてこれらの目的に特化

した堅牢な施設及び運用形態を提供するデータセンタあるいはインターネットデータセンタ（以下、iDCという）を利用するのが一般的である。

【0008】

また、サービス提供者は、サービスの対価を得るために、ユーザの使用量に応じて課金する機能やユーザ情報の管理機能を持つ必要がある。したがって、サービス提供者は、ユーザに利用させるアプリケーションと連携して動作可能な課金管理機能及びユーザ管理機能を開発し、運用する必要がある。

【0009】

【発明が解決しようとする課題】

サービス提供者は、上記iDCを利用することにより物理的なインフラや運用管理を利用可能である。しかしながら、サービス提供者の運営にあたって必要な機能には、iDCを利用するのみでは実現されない機能も多数ある。

【0010】

例えば上述したアプリケーションの使用量に応じたユーザへの課金管理機能、請求書の作成を支援する機能、請求書の発送を支援する機能、ユーザからの問い合わせに対する返答を支援する機能等は、iDCの利用のみでは十分でない。

【0011】

また、サービス提供者が他のサービス提供者と連携してユーザに機能を提供することで、ユーザの利用可能な機能を多くし、ユーザの利用数増加を試みる場合がある。この場合、双方のサービス提供者間で統一的なユーザ管理を行わなければならない。したがって、複数のサービス提供者間でユーザ管理を統合する機能が必要となる。

【0012】

これらの機能は、サービス提供者がサービスを提供していくために必要な機能であるが、サービス提供者はこれらの機能を提供可能なレベルの技術力を持たない場合がある。

【0013】

このような場合、サービス提供者がこれらの各種機能を自力で用意することはサービス提供にあたっての障害となる。

【0014】

そこで、iDCがこれらの各種機能を用意しておき、サービス提供者に提供するかわりに対価を得るという事業が成立する。このとき、iDCとしては、(1)サービス提供者毎に使用したい機能が異なるがそれへの対応が容易であること、(2)サービス提供者との契約内容に変更が生じた場合にその変更を上記事業に反映させることができること、(3)iDCが提供する各種機能の種類、内容に追加／変更がある場合、変更される機能の追加が容易であること、が実現できていなければ事業としての運用・保守にコストがかかってしまい、利益が出せないという問題がある。

【0015】

本発明は、以上のような実情に鑑みてなされたもので、サービス提供者、例えばASPを支援するためのコンピュータシステムによる情報処理サービスの提供方法、プログラム、情報処理サービス提供システムを提供することを目的とする。

【0016】

【課題を解決するための手段】

本発明を実現するにあたって講じた具体的手段について以下に説明する。

【0017】

第1の発明から第8までの発明は、コンピュータシステムによる情報処理サービスの提供方法に関する。

【0018】

第1の発明は、ユーザにネットワークを経由して提供されるサービスに対して付加価値サービスを提供するためのイベントを検出するステップと、イベントが検出された場合に、付加価値サービスを提供するデータ処理手段を呼び出すために用意されたライブラリを参照するステップと、ライブラリの参照結果を利用して付加価値サービスを提供するデータ処理手段に処理実行を依頼するステップからなる。

【0019】

これにより、サービスと付加価値サービスとを連携させることができる。した

がって、サービス提供者のみでは十分なサービスを提供できない場合であっても、付加価値サービスを利用してサービスの価値を高めることができる。また、サービス提供者は、付加価値サービスの運営を他の業者に依存することでサービス提供者としての管理業務を簡易化できる。

【0020】

なお、処理実行を依頼する側のサービスと依頼される側の付加価値サービスとが、別個のハードウェア上で実施される場合には、ライブラリはネットワーク経由で利用可能とする。

【0021】

データ処理手段には、例えばプログラム、コンピュータ・ハードウェア、オブジェクト、モジュール、プロセスなどがある。

【0022】

また、データ処理手段に対する処理実行の依頼方法としては、例えばデータ処理手段に対してリクエストを出力する手法、データ処理手段に対してデータを出力する手法、データ処理手段の処理を呼び出す手法などがある。

【0023】

第2の発明は、サービスの運用状況に関するデータを記憶手段に記憶するステップと、サービスに対して付加価値サービスを提供するデータ処理手段に対して、記憶手段に記憶された運用状況に関するデータを出力するステップとからなる。

【0024】

これにより、運用状況に関するデータを利用する各種サービス（例えば運用管理サービス、ユーザ管理サービス、課金管理サービスなど）を、付加価値サービスとして実施できる。

【0025】

この付加価値サービスをiDC業者が実施すれば、自データセンタの付加価値を高めることができる。

【0026】

また、複数のサービスに対するユーザ管理サービス、課金管理サービスを一つ

の付加価値サービス業者が請け負うことにより、ユーザ管理や課金管理を複数のサービス提供者間で容易に共通化できる。このため、ユーザは、複数のサービスへのシングルサインオン環境、課金請求の統合などのサービスを受けることができる。

【0027】

第3の発明は、サービスへのリクエストを取得するステップと、取得されたリクエストに応じて、そのサービスに対して付加価値サービスを提供するデータ処理手段に処理実行を依頼するステップとからなる。

【0028】

この第3の発明では、ユーザからサービスへのリクエストを一旦横取りし、この横取りしたリクエストに対応する付加価値サービスの実施を依頼する。

【0029】

第4の発明は、サービスの処理に基づいて発生する付加価値サービスのリクエストを取得するステップと、取得されたリクエストに応じて、そのサービスに対して付加価値サービスを提供するデータ処理手段に処理実行を依頼するステップとからなる情報処理サービスの提供方法である。

【0030】

この第4の発明では、サービスから付加価値サービスを利用するためのリクエストを横取りし、付加価値サービスを呼び出す。これにより、例えばサービスが付加価値サービスを利用する場合に必要な料金を、課金管理を行う付加価値サービスで課金することができる。

【0031】

第5の発明は、上記第3又は第4の発明と同様の情報処理サービスの提供方法である。この第5の発明は、リクエストの内容と処理実行依頼先とを対応付けたテーブルを参照し、取得されたリクエストの内容に対応するデータ処理手段に処理実行を依頼することを特徴とする情報処理サービスの提供方法である。

【0032】

このようなテーブルを利用することで、リクエストの内容に応じて適切な付加価値サービスの実行を依頼できるだけでなく、課題の最後に挙げた3つの点も実

現でき、iDCとしての事業運用が容易になる。

【0033】

第6の発明は、上記第1乃至第5までの発明と同様の情報処理サービスの提供方法である。この第6の発明では、データ処理手段は、複数のサービスに共同で利用される。

【0034】

したがって、複数のサービス提供者間で上述したユーザ管理や課金管理を共通化できる。また、複数のサービス提供者についてシングルサインオン環境をユーザに提供できる。また、ユーザに対して課金請求を統合できる。

【0035】

第7の発明は、上記第1乃至第6までの発明と同様の情報処理サービスの提供方法である。この第7の発明では、複数のデータ処理手段が連携して動作することでサービス提供者に対して付加価値サービスを提供する。

【0036】

これにより、例えば付加価値サービス業者が他の付加価値サービス業者に対して所定のサービスを依頼することもできる。したがって、各付加価値サービス業者間で連携しつつサービスの要求する付加価値サービス群を実施できる。

【0037】

第8の発明は、上記第1乃至第7までの発明と同様の情報処理サービスの提供方法である。この第8の発明では、サービス提供者毎に、データ処理手段の使用状況に応じた費用を費用計算手段によって計算するステップを付加している。

【0038】

これにより、付加価値サービスに対する対価をサービス提供者から得ることができる。

【0039】

なお、上記第1乃至第8の発明で実施される工程を実現可能なプログラム又はプログラムを記録したコンピュータ読み取り可能な記録媒体を、発明の対象としてもよい。

【0040】

このプログラム又はこのプログラムを記録した記録媒体を用いることによって、計算機システム、サーバやクライアント等の計算機に対して、簡単に上述した動作を実施可能な機能を付加することができる。

【0041】

また、上記第1乃至第8の発明で実施される工程を実現可能な手段を具備した計算機システムを発明の対象としてもよい。

【0042】

【発明の実施の形態】

以下、図面を参照しながら本発明の実施の形態について説明する。なお、以下に示す各図において、同一の部分については同一の符号を付してその説明を省略し、異なる部分についてのみ詳しく説明する。

【0043】

(第1の実施の形態)

本実施の形態においては、サービスをASPのアプリケーションサービスとし、このASPに対して付加価値サービスを提供する情報処理サービス提供システムについて説明する。

【0044】

なお、付加価値サービスとは、ASPがASPとしての業務を行うために必要とする共通のIT(情報技術)サービスである。また、付加価値サービスとは、ASPの運用管理を行うために必要となるサービスでもある。

【0045】

図1は、本実施の形態に係る情報処理サービス提供システムを例示するブロック図である。

【0046】

情報処理サービス提供システム1のユーザ(エンドユーザ及びユーザ企業など)2は、自己の操作するクライアント3からインターネット4を経由してASPサーバ5a又は5bにリクエストを出力する。このリクエストに基づいてASPサーバ5aのアプリケーション6a又はASPサーバ5bのアプリケーション6bは稼動する。その結果、ユーザ2は、ASPサービスを受ける。

【0047】

A S P サーバ 5 a、 5 b のイベント検出機能 7 は、 付加価値サービスの実施を依頼するイベントの発生を検出する。

【0048】

例えば、 A S P サーバ 5 a、 5 b が新規のユーザにアクセスされた場合、 ユーザを管理するために、 新規ユーザの登録作業を行う必要がある。そのため、 イベント検出機能 7 は、 新規のユーザからのアクセスを検出する。その他にも、 イベント検出機能 7 は、 ユーザの使用に応じて課金するためのイベント、 また A S P サーバ 5 a、 5 b の障害発生を検出する。

【0049】

ライブラリ 8 は、 付加価値サービスの呼び出しに利用されるデータ又はプログラムを格納している。ライブラリ 8 は、 A S P から付加価値サービスを呼び出すために、 付加価値サービスの A P I (Application Program Interface) を持つ。

【0050】

ライブラリ 8 は、 参照機能 9 を実現するハードウェアと同一のハードウェア上に備えられている必要はなく、 ネットワーク経由で参照機能 9 により参照可能とする。

【0051】

参照機能 9 は、 イベント検出機能 7 でイベントが検出された場合にライブラリ 8 を参照する。

【0052】

依頼機能 10 は、 参照機能 9 によるライブラリ 8 の参照結果を利用し、 イベント検出機能 7 で検出されたイベントの種別に応じた付加価値サービスを呼び出す。

【0053】

付加価値サービスは、 そのサービスを実現する機能（例えばプログラム、 プロセス、 モジュールなどで実現される機能）が処理実行を依頼されると実施される。付加価値サービスには、 例えばユーザ管理サービス 11 a、 課金管理サービス

11b、セキュリティサービス11cがある。この図1では、付加価値サーバ12aによってユーザ管理サービス11a及び課金管理サービス11bが実施される。また、付加価値サーバ12bによってセキュリティサービス11cが実施される。

【0054】

ASPサーバ5a、5bと付加価値サーバ12a、12bとは、ネットワーク4aを介して接続されている。

【0055】

なお、本実施の形態において、イベント検出機能7、参照機能9、ライブラリ8、依頼機能10はASPサーバ5a、5b側に配置されているが、例えば付加価値サーバ12a、12b側に配置してもよい。付加価値サーバ12a、12b側にイベント検出機能7が配置された場合、このイベント検出機能7はネットワーク4aを経由してASPサーバ5a、5bで発生するイベントを検出する。

【0056】

また、本実施の形態においては、ASPサーバ5a、5bと付加価値サーバ12a、12bとが異なるハードウェア上で稼動する場合を例として説明している。しかしながら、ASPサーバ5a、5bと付加価値サーバ12a、12bは同一のハードウェア上で稼動してもよい。

【0057】

図2は、本実施の形態に係る情報処理サービス提供システム1の変形例を示すブロック図である。

【0058】

ASPサーバ5cの記憶機能13は、ASPサーバ5cの運用状況データをデータベース14に記憶する。

【0059】

ASPサーバ5cの依頼機能15は、付加価値サービスを実施する機能に対してデータベース14に記憶されている運用状況データを出力する。

【0060】

付加価値サーバ12aで実施されるユーザ管理サービス11aと課金管理サー

ビス11b、及び付加価値サーバ12bで実施されるセキュリティサービス11cは、運用状況データを入力し、この運用状況データに基づいてそれぞれの処理を実行する。

【0061】

上記情報処理サービス提供システム1においては、ASPサービスと付加価値サービスとが連携して実施される。

【0062】

したがって、各種付加価値サービスをiDC業者が実施する場合、iDC業者は自己のデータセンタの付加価値を高めて事業を行うことができる。

【0063】

また、あるユーザ2が複数のASPを利用する場合、複数のASPが共通の付加価値サービスを利用することにより、このユーザに対するユーザ管理や課金管理を複数のASP間で統一化することができる。

【0064】

したがって、ユーザは、複数のASPに対するシングルサインオン環境を利用可能であり、課金請求を一本化できる。

【0065】

また、ASPは、本来ASPに利用する仕様で作成されていないアプリケーションであっても、ASPとして利用するために足りない機能を付加価値サービスで補充できる。

【0066】

したがって、容易にアプリケーションをASPとして利用可能である。また、各種サービスの充実によりユーザの利用を促進できる。

【0067】

なお、本実施の形態に係る情報処理サービス提供システム1の各種機能7～10、13、15は、記録媒体16に記録されたプログラムをコンピュータが読み取ることで実現可能とする。

【0068】

(第2の実施の形態)

本実施の形態においては、ASPサービスと付加価値サービスとを連携して実施する情報処理サービス提供システムについてさらに詳細に説明する。

【0069】

図3は、本実施の形態に係る情報処理サービス提供システムの連携状態を例示するブロック図である。

【0070】

ユーザ2は、ASP18のアプリケーションによるASPサービス18a～18hを利用する。

【0071】

ASPポータル17は、ユーザ2とASP18とのマッチメイク、ASP18に関するマーケティングを行う。さらに、ASPポータル17は、ASP間の連携とそのためのITサービス、ユーザに対するASP導入のためのコンサルタント、BPR (Business Process Re-engineering) 支援サービスを提供する。

【0072】

ASP18は、アプリケーションによるASPサービス18a～18gをユーザ2に提供する。

【0073】

付加価値サービス業者19は、複数のASPに対する共通のITサービス、ASPの運用管理サービスなどのような付加価値サービス11a～11jを提供する。

【0074】

付加価値サービスの一種である共通運用管理サービス11jは、付加価値サービス業者19が他の付加価値サービス業者の提供する付加価値サービス又は他のASPの提供するASPサービスなどを利用するための管理を行う。

【0075】

iDC業者20は、物理インフラ及び運用管理サービスを提供し、ホスティング(サーバ貸し)、ハウジング(場所貸し)を行う。

【0076】

ASP18によって提供されるASPサービス18a～18gは、必要に応じ

て付加価値サービス業者19によって提供される付加価値サービス11a～11j及びiDC業者20のiDC21を利用する。

【0077】

iDC21上には、付加価値サービス11a～11jを提供するためのアプリケーションプラットフォーム22が用意されている。

【0078】

図4は、本実施の形態に係る情報処理サービス提供システムの構成を例示するブロック図である。

【0079】

本実施の形態では、付加価値サービスをその機能に基づいて、ベーシックサービス、コミュニケーションサービス、ユーティリティサービス、バックエンドサービスの4種に分類して扱う。

【0080】

ベーシックサービスは、ASPサービスを提供するための基本機能を提供するサービスである。ベーシックサービスの例としては、ユーザ管理サービス11a、課金管理サービス11b、セキュリティサービス11c、SLM (Service Level Management) サービス11dなどがある。なお、SLMとは、ASPがユーザに対してサービスの質（稼働率やレスポンス）を保証するためのシステム管理機能と、このシステム管理機能を利用したサービスレベル保証をいう。

【0081】

ユーティリティサービスは、ASPサービスを容易に立ち上げるためのサービスである。ASPは、自己のサービスに対してユーティリティサービスを必要に応じて組みこむことで、ASP自身で提供することが困難なサービスであっても簡単に提供可能となる。ユーティリティサービスの例としては、情報分析サービス、情報提供サービス、印刷サービスなどがある。

【0082】

コミュニケーションサービスは、ユーザ2とASP18との間のコミュニケーションを確立するためのサービスである。コミュニケーションサービスの例としては、ポータルサービス、ユーザ2の電子メールの利用を可能とする電子メール

サービス11kなどがある。

【0083】

バックエンドサービスは、ASPサービスの運用のためにバックエンドで処理を行うサービスである。バックエンドサービスの例としては、EDI (Electronic Data Interchange) / EAI (Enterprise Application Integration) サービス、ジョブコントロールサービス11hなどがある。

【0084】

なお、EDIとは、商取引に関する情報を標準的な書式に統一して、企業間で電子的に交換する仕組みである。この仕組みでは、受発注や見積もり、決済、出入荷などに関わるデータを予め定められた形式にしたがって電子化し、専用線やVANなどのネットワークを通じて送受信する。

【0085】

また、EAIとは、企業内で業務に使用される複数のコンピュータシステムを有機的に連携させ、データやプロセスの効率的な統合をはかること、またそれを支援する一連の技術やソフトウェアの総称である。

【0086】

さらに、本実施の形態においては、ASPサービスと付加価値サービスとの間の連携方法に基づいて、付加価値サービスをユーティリティ連携サービス、ベーシック連携サービス、バックエンド連携サービスの3種に分類して扱う。

【0087】

ユーティリティ連携サービスは、ユーザ又はASPから必要に応じて呼び出されるサービスであり、主に上記のユーティリティサービス及びコミュニケーションサービスが含まれる。ユーティリティ連携サービスの呼び出しは、ユーティリティ連携サービスにアクセスするルーチンをASPの内部に組みこむことで実現する。

【0088】

ベーシック連携サービスは、ユーザがASPにアクセスする際に介在するサービスである。ユーザの認証を行うユーザ管理サービス11a、ASPによるユーザに対する課金管理サービス11bなどのような上記ベーシックサービスは、ベ

ー・シック連携サービスに該当する。ベーシック連携サービスは、ユーザの操作するクライアントからASPに対するサービス実行リクエストが発生する度に自動的に起動される。

【0089】

加えて、このベーシック連携サービスには、ASPサービスから付加価値サービスを利用する際に介在するサービスも含まれる。付加価値サービス業者からASPに対する課金管理などのような上記ベーシックサービスは、ベーシック連携サービスに該当する。ベーシック連携サービスは、ASPサービスが付加価値サービスを呼び出す際に自動的に起動される。

【0090】

バックエンド連携サービスは、ASP自体の稼動状態を監視・制御するサービスである。上記バックエンドサービスは、このバックエンド連携サービスに該当する。バックエンド連携サービスは、付加価値サービス業者側からASPに対して処理の起動・停止などの操作を行う。

【0091】

このようなユーティリティ連携サービス、ベーシック連携サービス、バックエンド連携サービスとASPサービスとの間の連携を実現するために必要な3種類のインターフェースについて説明する。

【0092】

各種サービスは、それぞれに対応するプログラム又はハードウェアにリクエストが出力された場合、あるいはそれぞれに対応するプログラム又はハードウェアを呼び出すことにより実行される。

【0093】

ユーティリティ連携サービスとASPサービスの連携、ベーシック連携サービスとASPサービスの連携、バックエンド連携サービスとASPサービスとの連携は、以下のアーキテクチャで実現する。

【0094】

ユーティリティ連携サービスとASPサービスとの間の連携を可能とするために、情報処理サービス提供システムは、ASPサービスからユーティリティ連携

サービスを呼び出すためのAPIをライブラリ8の形で備える。そして、ASPサービスを提供する時点で、ライブラリ8を利用してASPサービス側から付加価値サービスを呼び出すことで、ユーティリティ連携サービスとASPサービスとの間の連携が可能となる。

【0095】

なお、呼び出し側のASPサービスと呼び出される側のユーティリティ連携サービスとが別のハードウェア上で稼動する場合があるため、本実施の形態においても、上記第1の実施の形態の場合と同様に、ライブラリ8をネットワーク経由で利用可能とすることが望ましい。

【0096】

ベーシック連携サービスとASPサービスとの間の連携を可能とするために、情報処理サービス提供システムは、ユーザの操作するクライアントからのリクエストを受け付けるネットワークサーバ（例えばWebサーバなどのプロセス）23の内部に、ベーシック連携サービスの実行を依頼するユーザ側リクエストフック機能24を埋め込むことで連携を可能とする。

【0097】

この場合の処理の流れは、まずネットワークサーバ23がユーザからのリクエストを受け取り、ユーザ側リクエストフック機能24のリクエスト取得機能24aがリクエストを横取りし、依頼機能24bがベーシック連携サービスに処理を依頼する。これにより、一旦ベーシック連携サービスに制御が移る。そして、ユーザ認証などの必要な機能を実行し、ASPのアプリケーションに制御が移る。

【0098】

また、情報処理サービス提供システムは、ASPサービスのアプリケーションからユーティリティ連携サービスに対するリクエストをリクエスト取得機能25aで横取りし、依頼機能25bでベーシック連携サービスに処理を依頼するアプリケーション側リクエストフック機能25を備える。

【0099】

バックエンド連携サービスとASPサービスとの間の連携を可能とするために、情報処理サービス提供システムは、ASPサービスを実現するアプリケーショ

ンのプロセス状態を定常的に監視し、その状態に応じてバックエンド連携サービスからASPサービスを実現するアプリケーションのプロセスを操作し、連携を可能とする。なお、ASPサービスが独立したプロセスでなく、アプリケーションサーバ上で動作するスレッドの場合には、アプリケーションサーバの持つ監視機能をバックエンド連携サービスから呼び出す。

【0100】

上記のような連携のためのアーキテクチャを実現するためのリクエストフック機能について以下で詳細に説明する。

【0101】

上述したように、ユーザ側リクエストフック機能24とアプリケーション側リクエストフック機能25からなる2種類のリクエストフック機能は、リクエストを一旦横取りして別の処理の実行を依頼する。

【0102】

図5は、ユーザ側リクエストフック機能24による連携状態を例示するブロック図である。

【0103】

ユーザ側リクエストフック機能24は、ユーザ2からASPに対するリクエストを横取りし、ベーシック連携サービスを呼び出す。この呼び出しにより、各種ASPサービスを利用するユーザ2が識別され、ユーザ認証及び課金処理が代行される。

【0104】

すなわち、このユーザ側リクエストフック機能24は、ユーザ2からのASPサービスの利用に介在し、ユーザ認証や与信・課金などのユーザ管理機能の実行を依頼する。

【0105】

図6は、アプリケーション側リクエストフック機能25による連携状態を例示するブロック図である。

【0106】

アプリケーション側リクエストフック機能25は、ASPサービスからユーテ

イリティ連携サービスを利用する旨のリクエストを横取りし、ベーシック連携サービスを呼び出す。この呼び出しにより、ASPが付加価値サービスを利用した量に応じて、ASPに対して適正に課金を行うことが可能になる。

【0107】

すなわち、このアプリケーション側リクエストフック機能25は、ASPによる付加価値サービスの利用に介在し、認証や与信・課金などの管理処理の実行を依頼する。

【0108】

以下に、リクエストフック機能24、25の実現方法を説明する。なお、以下においてはユーザ2からWebサーバを経由してリクエストが取得される場合について説明する。しかしながら、リクエストフック機能は、ユーザからのアクセスを受け付けるあらゆる種類のネットワークサーバに同様に適用可能である。

【0109】

付加価値サービスとしては、iDC21上で動作する1以上のASPを利用するユーザに関する情報を統合して管理するサービス、1以上のASPの課金情報を統合して管理するサービス、1以上のASPに関する稼動状況を一元管理するサービス、1以上の付加価値サービスに関する稼動状況を一元管理するサービスがある。

【0110】

リクエストフック機能24、25は、ASPを呼び出すリクエスト又はユーザイリティ連携サービスを呼び出すリクエストを横取りし、リクエストに関する情報をベーシック連携サービスに出力する。これにより、上記の付加価値サービスが実施される。

【0111】

ユーザがWebサーバを経由して利用するASPサービスの一例としては、建設ASPサービス18hがある。建設ASPサービス18hとは、あるビルの建設のために短期間のみ結成されるJV（ジョイントベンチャー）を対象とするASPサービスである。JVはある期間が経過すると解散する。したがって、JVにとっては、会社としての機能を果たすために必要なアプリケーション（会計シ

システムやCADソフトなど)を自分たちで購入して保守・運用するよりも建設ASPサービス18hを利用する方が効率的となる。建設ASPサービス18hは、JVが必要とするアプリケーションを提供することに特化されている。

【0112】

ベーシック連携サービスには、ユーザ管理サービス11a、課金管理サービス(予信・課金サービス)11bがある。ユーティリティ連携サービスには、電子メールサービス11kがある。

【0113】

ユーザ管理サービス11aは、指定されたユーザに対するパスワードにより認証し、またユーザに要求されたデータへのアクセス権を確認するサービスである。

【0114】

課金管理サービス11bは、指定されたユーザの予信状況(使用料の上限に達していないか等)を調査し、課金用の情報を保持して月締めで集計するサービスである。

【0115】

電子メールサービス11kは、大量の送付先に広告用のメールを送り、その中に各送付先に特化した情報(宛先ユーザの名前など)を織り込むサービスである。

【0116】

図7は、リクエストフック機能24、25を含む情報処理サービス提供システムのアーキテクチャを例示するブロック図である。

【0117】

また、図8は、情報処理サービス提供システムのアーキテクチャと具体的なサービスとの関係を例示したブロック図である。

【0118】

ユーザ側リクエストフック機能24とアプリケーション側リクエストフック機能25とは、機能的に似た点が多いため共通部分を部品化できる。以下で、この部品をリクエストフックフレームワーク26とする。

【0119】

リクエストフックフレームワーク26はJavaのようなプログラミング言語で開発される。リクエストフックフレームワーク26の利用する対応テーブルは、設定ファイル等で定義される。この対応テーブルを利用することで、リクエストの種類（リクエストがどのサービスに対応するかなど）によって起動するベーシック連携サービスを切り替え可能となる。

【0120】

ユーザ側リクエストフック機能24は、Webサーバ28もしくはその前段のProxyサーバ（以下、「Gateサーバ」という）27の機能とする。この例では、Gateサーバ27により、ユーザ側リクエストフック機能24を実現している。

【0121】

ユーザはWebブラウザ29からWebサーバ28にアクセスを要求する。すると、Gateサーバ27によるユーザ側リクエストフック機能24は、Webブラウザ29からWebサーバ28へのリクエストを横取りする。そして、ユーザ側リクエストフック機能24は、ベーシックサービスラッパ30（例えばユーザ管理ラッパ30a、課金管理ラッパ30b）を用いて、ベーシック連携サービス（プロセス）31（例えばユーザ管理サービス31a、課金管理サービス31b）に実行を依頼する。

【0122】

この結果、アクセスしたユーザが特定され、その課金情報が蓄えられ、アクセスしたユーザがアプリケーションサーバ32上のASP33（例えば建設ASP33a）を利用可能か否かが判断される。

【0123】

ユーザのASP33の利用が許可されると、ユーザはASP33を利用する。さらに、このASP33に加えてユーザがユーティリティ連携サービス（プロセス）35（例えば電子メールサービス35a）を利用する場合、アプリケーション側リクエストフック機能25は、ASP33からユーティリティ連携サービス35へのリクエストを横取りする。

【0124】

そして、アプリケーション側リクエストフック機能25は、ベーシックサービスラッパ30を用いて、ベーシック連携サービス31に実行を依頼する。

【0125】

この結果、ユーザが特定され、その課金情報が蓄えられ、ユーザがユーティリティ連携サービス35を利用可能か否かが判断される。

【0126】

ユーティリティ連携サービス35の利用が許可されると、ASP33はユーティリティサービスラッパ34（例えば電子メールサービスラッパ34a）を利用し、ユーティリティ連携サービス35に実行を依頼する。

【0127】

以上のような連携により、ユーザ管理サービス11a、課金管理サービス11bが実行される。

【0128】

この結果、建設ASP33aは、本来サービスとして備えていない電子メールサービス11kをユーザに提供でき、自分のサービスに付加価値をつけることができる。

【0129】

付加価値サービス業者は、電子メールサービス11kを建設ASP33aに提供した対価としての課金情報を蓄え、建設ASP33aに請求することができる。

【0130】

図9は、リクエストフック機能24、25の構造を例示するクラス図である。この図は設計表記法の一つであるUML (Unified Modeling Language) の形式で書かれている。

【0131】

また、図10は、リクエストフック機能24、25の構造と具体的なサービスとの関係を例示したクラス図である。

【0132】

矩形及び3分割された矩形は、クラスを表している。矩形の中にかかれた名前（例えば「Gateサーバ」）はこのクラスの名前である。3分割された矩形はクラスのより詳細な表記法である。矩形の最上部の中身（例えば「フックライブラリ」）はクラス名を、矩形の最下部の中身（例えば「前処理()」、「後処理()」）はこのクラスが受け付けることのできる処理要求を示している。なお、処理要求とは、オブジェクト指向の場合にはメソッドに該当する。

【0133】

破線の矢印は、根元側のクラスが先端側のクラスにアクセスして処理を呼び出したり情報を得たりすることを表している。

【0134】

フックライブラリ37と、ベーシックサービスラッパ規約38に準拠し、処理メソッドを持ち、フックライブラリ37から呼び出されるクラス30（例えばユーザ管理ラッパ30a、課金管理ラッパ30b）とは、ユーザ側リクエストフック機能24とアプリケーション側リクエストフック機能25に共通の機能である。

【0135】

フックライブラリ37は、リクエストフック機能24、25の入り口になるクラスである。ユーザ側リクエストフック機能24の場合、Gateサーバ27はフックライブラリ37を呼び出し、ユーザ側リクエストフック機能24を実行する。

【0136】

一方、アプリケーション側リクエストフック機能25の場合、ASP33（例えば建設ASP33a）がユーティリティ連携サービス35（例えば電子メールサービス35a）を呼び出す際に使用されるユーティリティサービスラッパ規約39に準拠したユーティリティサービスラッパ34（例えば電子メールサービスラッパ34a）がフックライブラリ37を呼び出す。

【0137】

リクエストフック機能24、25を構成する各クラスの役割を以下に示す。

【0138】

Gateサーバ27は、クライアント29からASP33に対するリクエストを仲介するProxyサーバである。Gateサーバ27は、リクエストを転送する前後にフックライブラリ37を呼び出す。

【0139】

ASP33（例えば建設ASP33a）は、ASPサービスの実体である。ASP33は、サービスの実行中にユーティリティ連携サービス35（例えば電子メールサービス35a）を利用する。

【0140】

フックライブラリ37は、フック機構の入り口となる。このフックライブラリ37は、ASP33やユーティリティ連携サービス35の実行前に呼び出される前処理メソッドと、実行後に呼び出される後処理メソッド、及びラッパに渡す情報を受け取るための情報設定メソッドを持つ。

【0141】

前処置メソッド及び後処理メソッドは、サービスーラッパ対応付け部40の内容にしたがって、ベーシックサービスラッパ規約38に準拠したベーシックサービスラッパ30を呼び出す。

【0142】

ベーシックサービスラッパ規約38は、リクエストフック機能24、25で起動されるベーシック連携サービス31をフックライブラリ37から起動するラッパオブジェクトが準拠すべき規約である。

【0143】

具体的には、ベーシックサービスラッパ規約38においては、処理メソッド及び情報設定メソッドを持つことが定められている。この規約を用意することにより、この規約に準拠しているラッパであればベーシックサービスラッパとして任意のものを使用可能であり、iDCとしてはASPに提供するベーシックサービスを隨時変更していくことができるようになる。

【0144】

ベーシックサービスラッパ30（例えば予信・課金ラッパ、ユーザ管理ラッパ）は、ベーシックサービス規約に準拠したクラスである。

【0145】

ベーシック連携サービス31（例えばユーザ管理サービス31a、課金管理サービス31b）は、ユーザ認証、課金、実行状況ログの記憶などのようなサービスの実体である。

【0146】

ユーティリティサービスラッパ規約39は、ASP33から利用されるユーティリティサービスラッパ34が準拠すべき規約である。

【0147】

ユーティリティサービスラッパ34（例えば電子メールサービスラッパ34a）は、ASP33からの利用要求をユーティリティ連携サービス35に転送する役割を果たすクラスである。このクラスでは、転送の前後にフックライブラリ37の前処理メソッド、後処理メソッドを呼び出す。

【0148】

ユーティリティ連携サービス35（例えば電子メールサービス35a）は、ASP33から利用されるサービスの実体である。

【0149】

図11は、ユーザ側リクエストフック機能24の前段の動作を例示するシーケンス図である。この図11は、ユーザが建設ASP33aを使用するまでの処理について示している。

【0150】

また、図12は、ユーザ側リクエストフック機能24の後段の動作を例示するシーケンス図である。この図12は、ユーザが建設ASP33aの使用を終えた後の処理について示している。

【0151】

ユーザのWebブラウザ29によって建設ASP33aが利用される前に、ユーザ管理サービス31aはユーザの認証チェックを行い、課金管理サービス31bはユーザのASPサービス使用料のチェックを行う。

【0152】

また、ユーザによって建設ASP33aが利用された後に、課金管理サービス

はユーザの課金情報を蓄積する。

【0153】

シーケンス図中の実線の矢印はメソッドの呼出を、破線の矢印は呼び出された処理を終えて制御が戻ることを示している。両種別の矢印には、必要に応じて呼び出されるメソッド名（実線の矢印の場合、引数を（）で囲っている）、あるいは返される値が矢印上部に記述されている。返される値も（）で囲っている。フックライブラリ37を実現する上で重要な働きを持つメソッドの呼び出しは太線で強調している。

【0154】

まず、図11に示すように、ユーザは、Webブラウザ29などのクライアントを用いて建設ASP33aの使用を要求する。この時点でユーザはWebサーバ28にアクセスしておらず、実際にはGateサーバ27がこのリクエストを受け付けている。

【0155】

次に、Gateサーバ27は、ユーザによって指定されたURLに基づいて、要求されているサービスが何かを把握する検索処理を実行する。この検索処理は、サービス名-URL対応付け部41に依頼する。

【0156】

表1は、サービス名-URL対応付け部41が使用する対応テーブルを例示している。

【0157】

【表1】

表 1

サービス名	URL
ASPトップ	/ASP/ASP.jsp
課金情報閲覧	/ASP/accounting.jsp
付帯	/ASP/option.jsp
システム連携1	/ASP/trans.jsp
システム連携2	/servlet/asp.transSvr
建設ASP	/JV/jv.jsp

【0158】

この表1を参照することで、URLから「建設ASP」などのようなサービス名を得ることができる。

【0159】

次に、Gateサーバ27は、このようにして得たサービス名及びユーザが入力したユーザ名などの情報をフックライブラリ37に渡す。これらの情報は、後にフックライブラリ37から各ラッパに渡され、ベーシック連携サービスが処理を実行するための引数として使われる。

【0160】

表2は、ユーザ側リクエストフック機能24が引数としてベーシック連携サービスに渡す値の例である。

【0161】

【表2】

表 2

項目名	値
利用者名	“Shigeru/Matsumoto”
パスワード	“1111111111”
呼出側サービス名	“建設 ASP”
呼出先サービス名	“Gate サーバ”
データサイズ	-
データ型	-
終了状態	-

【0162】

表3は、アプリケーション側リクエストフック機能25が引数としてベーシック連携サービスに渡す値の例である。

【0163】

【表3】

表 3

項目名	値
利用者名	“建設 ASP 管理者”
パスワード	“aaaaaa”
呼出側サービス名	“電子メール. メイル送信”
呼出先サービス名	“建設 ASP”
データサイズ	—
データ型	—
終了状態	—

【0164】

次に、Gate サーバ27は、フックライブラリ37の前処理メソッドを実行する。すると、フックライブラリ37は、サービスーラッパ対応付け部40に依頼して建設ASP33aの前処理に必要なラッパ群を取得する。

【0165】

表4は、サービスーラッパ対応付け部40が使用するテーブルを例示している

【0166】

【表4】

表 4

サービス	ラッパ	
建設 ASP. 前処理	ユーザ管理、課金管理	ユーザ側リクエスト フック機能用
建設 ASP. 後処理	課金管理	
システム連携 1. 前処理	ユーザ管理	
システム連携 2. 後処理	課金管理、 付加価値サービス利用ログ	
...	...	
電子メール. メール送信. 前処理	ユーザ管理、課金管理	アプリケーション側 リクエスト フック機能用
電子メール. メール送信. 後処理	課金管理	
電子メール. メール受信. 前処理	ユーザ管理、課金管理	
電子メール. メール受信. 後処理	課金管理、 付加価値サービス利用ログ	
...	...	

【0167】

具体的には、フックライブラリ37は、既に表2のような情報を受け取っているため、「建設ASP」をキーワードとして表4に対する検索を行う。

【0168】

これにより、フックライブラリ37は、建設ASP33aの前処理に必要なラッパとしてユーザ管理ラッパ30a、課金管理ラッパ30bを獲得する。

【0169】

次に、フックライブラリ37は、獲得したラッパを通じて実際のサービスを呼び出す。呼び出し順序は先にユーザ管理サービス31aを呼び出し、次に課金管理サービス31bを呼び出す。具体的には、フックライブラリ37は、先の処理で設定された表2の情報を獲得したラッパに渡し、各ラッパは備えている処理メソッドを呼び出す。

【0170】

ユーザ管理ラッパ30aは、ユーザ管理サービス31aに特化した操作メソッドを呼び出すとともに、ユーザ名とパスワードをユーザ管理サービス31aに渡す。

【0171】

すると、ユーザ管理ラッパ30aは、ユーザが正当であるか否かをチェックした結果をユーザ管理サービス31aから受け取る。その結果は、さらにユーザ管理ラッパ30aからフックライブラリ37に返される。

【0172】

チェック結果に問題がなければ、同様に課金管理ラッパ30bが課金管理サービス31bのメソッドを呼び出す。これにより、課金管理サービス31bは、例えばこのユーザの使用料が契約上の上限値を上回っていないか否かをチェックする予信管理を行う。

【0173】

これらの処理が終わると、制御はフックライブラリ37からGateサーバ27に戻る。

【0174】

Gateサーバ27は、Webサーバ28に対してユーザが指定したURLを通知し、建設ASP33aによるサービスが提供される（図11最大部、矩形で囲まれた部分）。このサービスとしては、例えばWebアプリケーションとして構築された資材の発注システムやスケジュールの管理ツールの利用がある。

【0175】

続いて、図12に示すように、Gateサーバ27は、情報設定メソッドを用いてフックライブラリ37から後処理で使うベーシック連携サービスに渡す情報を引き渡す。引き渡される情報には、例えば課金に使うためのこのユーザの接続時間やログを残すための時刻情報などがある。

【0176】

次に、Gateサーバ27は、フックライブラリ37の後処理メソッドを呼び出す。すると、前処理の場合と同様に、フックライブラリ37は、サービスラッパ対応付け部40から必要なラッパを取得し、該当するラッパの処理メソッドを呼び出す。これにより、ベーシック連携サービス（この例では課金管理サービス31b）が利用される。

【0177】

図13は、アプリケーション側リクエストフック機能25の前段の動作を例示するシーケンス図である。この図13は、ユーザが電子メールサービス35aを使用するまでの処理について示している。このようにサービスラッパ対応付け部40を用いることで、ASP毎に低脅威される付加価値サービスの種類を切り替えることができる。また、表4のようにテーブルで設定すれば、このテーブルを修正するだけでASPへ提供する付加価値サービスの種類を変更できるため、IDCとしては運用が容易になるだけでなく、ASPに提供する付加価値サービスの種類に柔軟性を持たせることができる。これは課題として前述した問題の解決策となっている。

【0178】

また、図14は、アプリケーション側リクエストフック機能25の後段の動作を例示するシーケンス図である。この図12は、ユーザが電子メールサービス35aの使用を終えた後の処理について示している。

【0179】

この図13及び図14は、すでに建設ASP33aが使われている時点から記述されている。また、図13及び図14は、ユーザからのメール送信のリクエストに応じるために電子メールサービス35aの使用が必要となった場合を例示している。

【0180】

建設ASP33aの処理の一部によってユーティリティ連携サービスの1つである電子メールサービス35aが利用される前に、ユーザ管理サービス31aは、利用元である建設ASP33aの認証チェックを行い、課金管理サービス31bは利用元である建設ASP33aの付加価値サービス使用料のチェックを行う。

【0181】

また、建設ASP33aの処理の一部によってユーティリティ連携サービスの1つである電子メールサービス35aが利用された後に、課金管理サービス31bは、利用元である建設ASP33aの課金情報を蓄積する。

【0182】

まず、図13に示すように、建設ASP33aは電子メールサービスラッパ34aを生成し、ラッパ34aに対してメール送信の要求する。

【0183】

すると、電子メールサービスラッパ34aは、フックライブラリ37に対して情報を渡す。

【0184】

フックライブラリ37は、前処理メソッドを呼び出し、実際の電子メールサービス35aに対してメール送信の処理を依頼し、図14に示すように後処理メソッドを呼び出す。

【0185】

フックライブラリ37は、ラッパを取得するためのキーワードとしてサービス名「電子メール」に加え、利用するメソッド名「メール送信」も使用する。その他のフックライブラリ37の動作は、ユーザ側リクエストフック機能24の場合

と同様である。ここでも、ユーザ側リクエストフック機能と同様に表4のようなサービスラッパ対応付けテーブルを使用する。これにより、ASPに提供する付加価値サービスの種類を固定することなく、より柔軟性を持たせることができるため、iDCはASPの要望にあったきめ細かな内容で付加価値サービスを提供できる。また、ASPとの契約内容に変更が生じた場合はこのテーブルを修正するだけでよく、ライブラリ自体の再コンパイルが不要となるためiDC側の運用コストを軽減することができる。

【0186】

なお、必要になった時点でそれぞれのラッパを初期化する方式の場合、ラッパの獲得要求からサービス提供までに処理時間がかかる。したがって、事前にラッパ群を生成してプリング（保留）しておき、処理の高速化を図る方法も考えられる。

【0187】

以上のような情報処理サービス提供システムを実現すると、iDC業者20、ASP18は以下のような利点を得ることができる。

【0188】

まず、iDC業者20は、iDC21上にアプリケーションプラットフォーム22を用意し、各種の付加価値サービスを提供することで、データセンタ単独での事業を行う場合よりも自データセンタの付加価値を高めることができる。

【0189】

ASP18は、自己の提供するASPサービスに付加価値サービスを連携させることができる。

【0190】

これにより、本来ASPサービスの提供用として作られていないアプリケーションをASPサービス用として機能させる場合に、課金管理やシステム監視などの不足しているサービスを付加価値サービスで補充できる。したがって、ソフトウェアのASP化が容易になる。

【0191】

また、ASP18は、他のASP業者との間でユーザ管理や課金管理を共通化

できるため、ユーザに対して複数のASPへのシングルサインオン環境を提供でき、課金請求を統合できる。複数のASPへのシングルサインオン環境や課金請求の一本化は、ユーザ2にとっても管理の一本化という利点となるため、ユーザ獲得を促進できる。

【0192】

なお、上記各実施の形態に係る情報処理サービス提供システムで説明した各機能、各要素は、同様の作用・機能を実現可能であれば配置を変更させてもよく、また各機能、各要素を自由に組み合わせてもよい。

【0193】

上記各実施の形態に係る情報処理サービス提供システムの各機能、各要素は、コンピュータに実行させることのできるプログラム、モジュール、プロセスとして提供できる。このプログラムにより上記各機能、各要素を提供する場合には、例えば磁気ディスク（フロッピーディスク、ハードディスク等）、光ディスク（CD-ROM、DVD等）、半導体メモリなどの記録媒体16にそのプログラムを書き込んでコンピュータに適用可能である。また、このプログラムは通信媒体により計算機、計算機システムに伝送しても適用可能である。コンピュータは、記録媒体16に記録されたプログラムを読み込み、そのプログラムによって動作が制御されることにより、上述した処理を実行する。

【0194】

【発明の効果】

以上詳記したように本発明においては、ASPサービスに対して付加価値サービスを連携させることができる。これにより、ASPサービスの実施を容易化できる。また、ASPサービスの内容を充実させることができる。

【図面の簡単な説明】

【図1】

本発明の第1の実施の形態に係る情報処理サービス提供システムを例示するブロック図。

【図2】

同実施の形態に係る情報処理サービス提供システムの変形例を示すブロック図

である。

【図3】

本発明の第2の実施の形態に係る情報処理サービス提供システムの連携状態を例示するブロック図。

【図4】

同実施の形態に係る情報処理サービス提供システムの構成を例示するブロック図。

【図5】

ユーザ側リクエストフック機能による連携状態を例示するブロック図。

【図6】

アプリケーション側リクエストフック機能25による連携状態を例示するブロック図。

【図7】

リクエストフック機能を含む情報処理サービス提供システムのアーキテクチャを例示するブロック図。

【図8】

同実施の形態に係る情報処理サービス提供システムのアーキテクチャと具体的なサービスとの関係を例示したブロック図。

【図9】

リクエストフック機能の構造を例示するクラス図。

【図10】

リクエストフック機能の構造と具体的なサービスとの関係を例示したクラス図

【図11】

ユーザ側リクエストフック機能の前段の動作を例示するシーケンス図。

【図12】

ユーザ側リクエストフック機能の後段の動作を例示するシーケンス図。

【図13】

アプリケーション側リクエストフック機能の前段の動作を例示するシーケンス

図。

【図14】

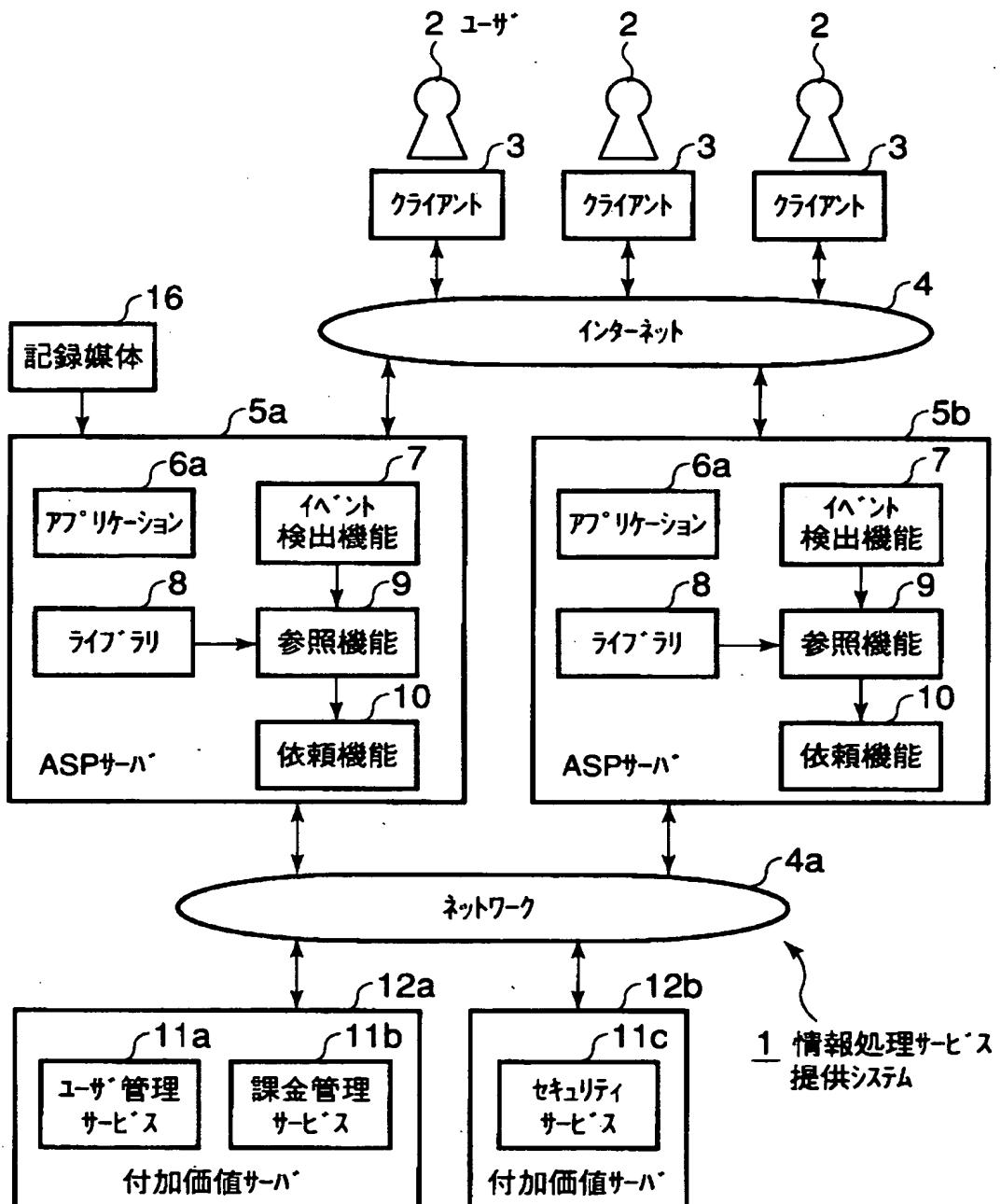
アプリケーション側リクエストフック機能の後段の動作を例示するシーケンス図。

【符号の説明】

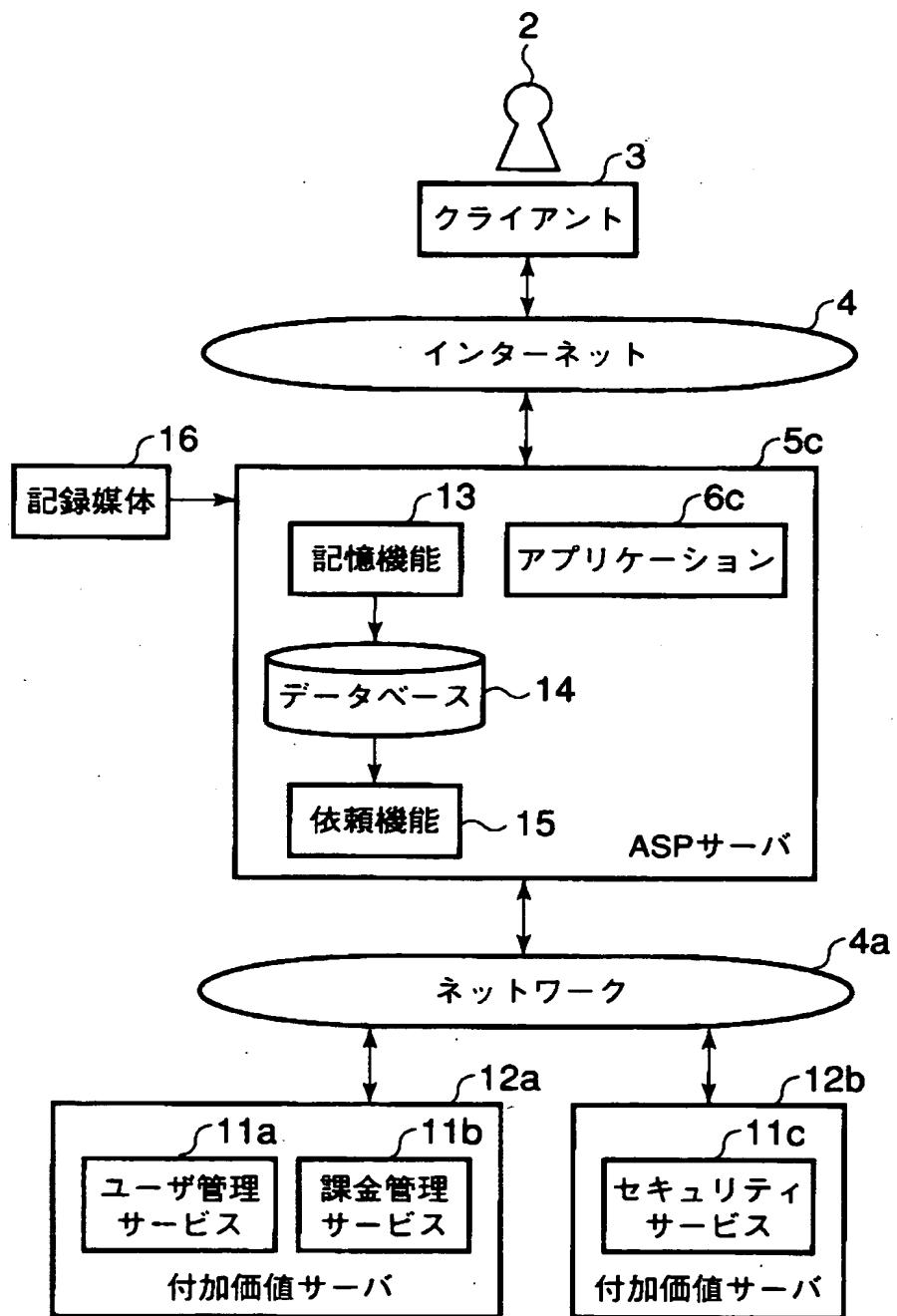
- 1 …情報処理サービス提供システム
- 2 …ユーザ
- 3 …クライアント
- 5 a、 5 b …ASPサーバ
- 6 a、 6 b …アプリケーション
- 7 …イベント検出機能
- 8 …ライブラリ
- 9 …参照機能
- 10、 15、 24 b、 25 b …依頼機能
- 12 a、 12 b …付加価値サーバ
- 13 …記憶機能
- 14 …データベース
- 17 …ASPポータル
- 18 …ASP
- 19 …付加価値サービス業者
- 20 …iDC業者
- 21 …iDC
- 22 …アプリケーションプラットフォーム
- 24 …ユーザ側リクエストフック機能
- 24 a、 25 a …リクエスト取得機能
- 25 …アプリケーション側リクエストフック機能
- 30 …ベーシックサービスラッパ
- 31 …ベーシック連携サービス
- 34 …ユーティリティサービスラッパ

- 3 5 …ユーティリティ連携サービス
- 3 7 …フックライブラリ
- 3 8 …ベーシックサービスラッパ規約
- 3 9 …ユーティリティサービスラッパ規約
- 4 0 …サービスーラッパ対応付け部
- 4 1 …サービス名-URL対応付け部

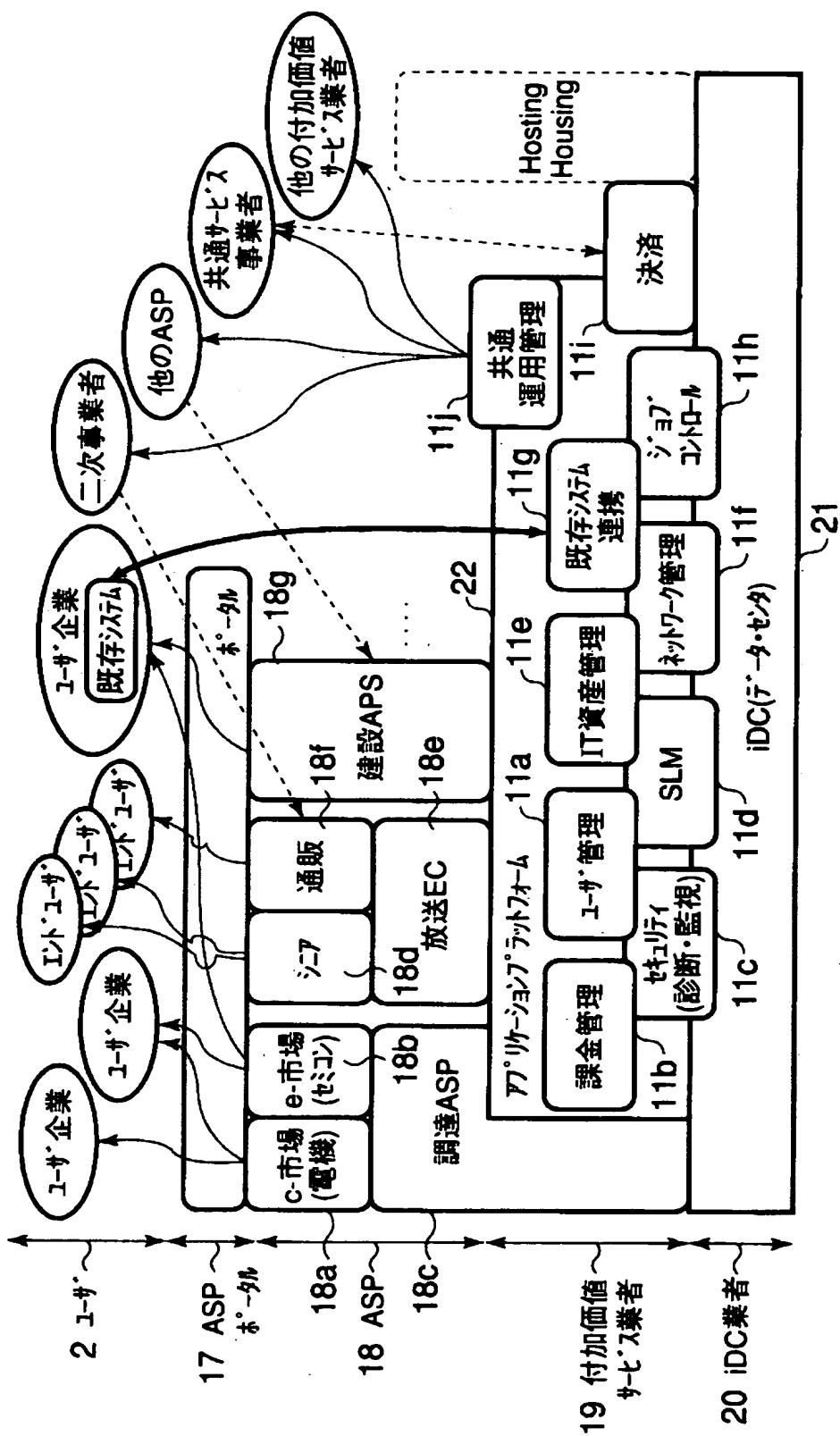
【書類名】 図面
【図 1】



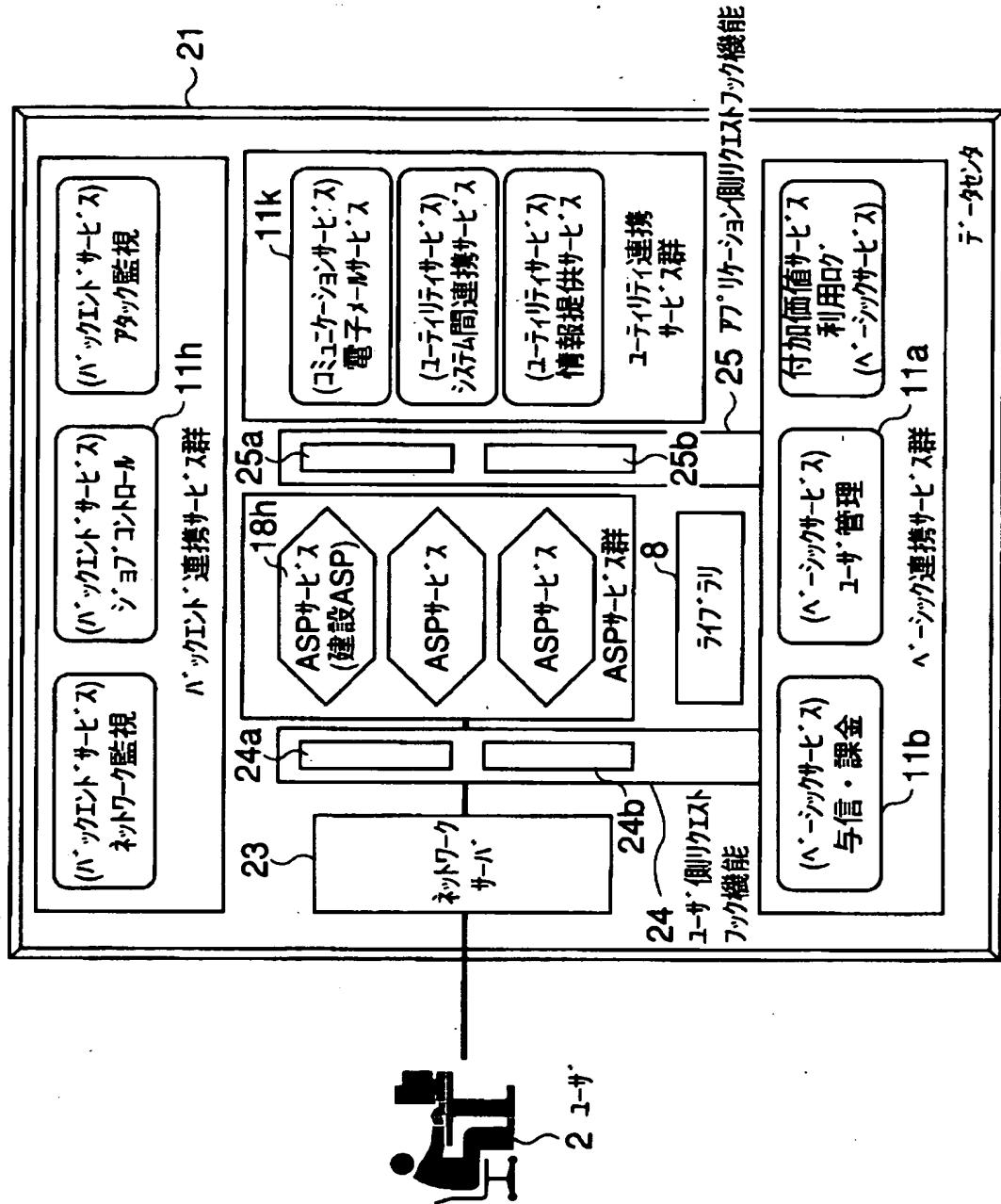
【図2】



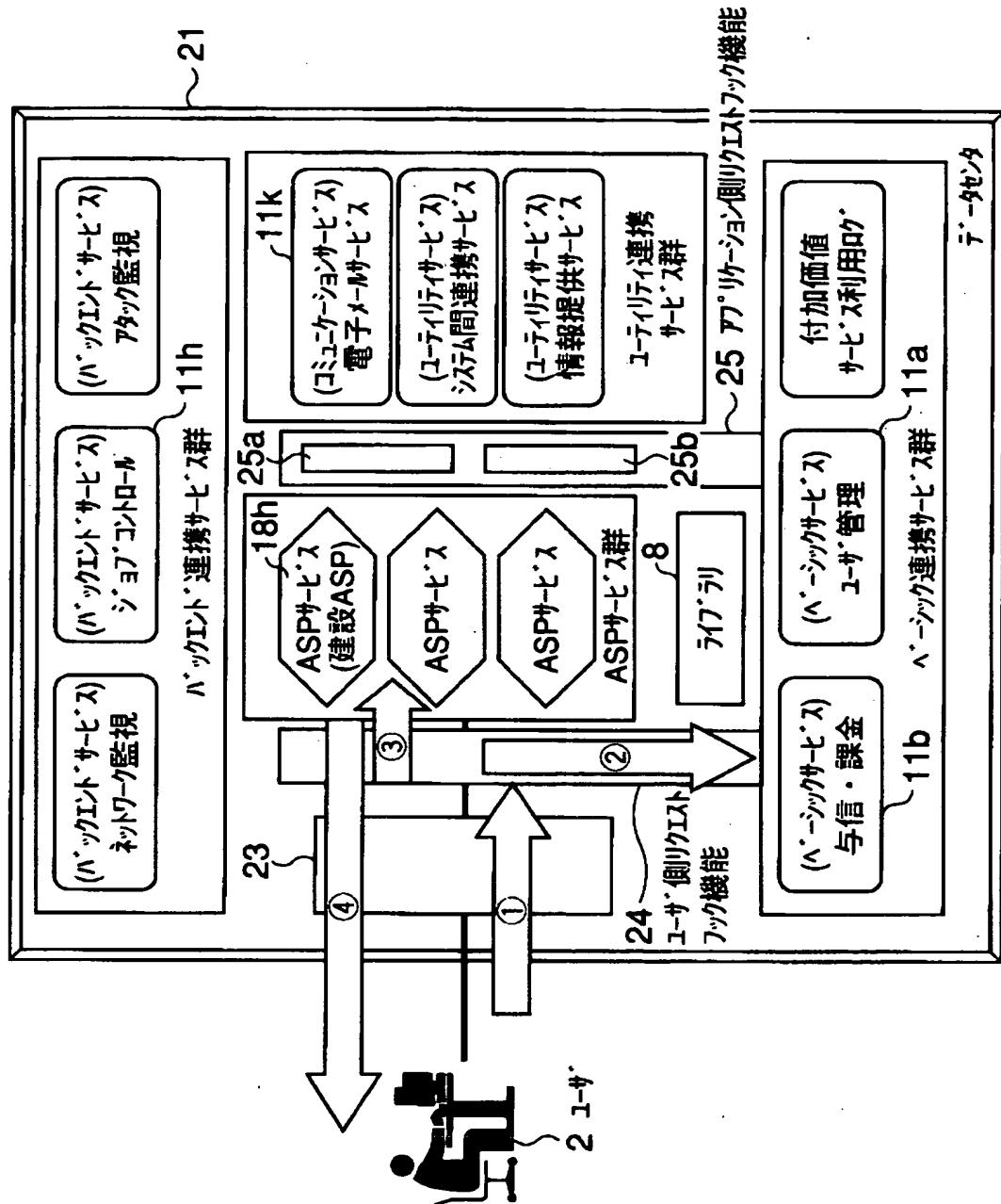
【図3】



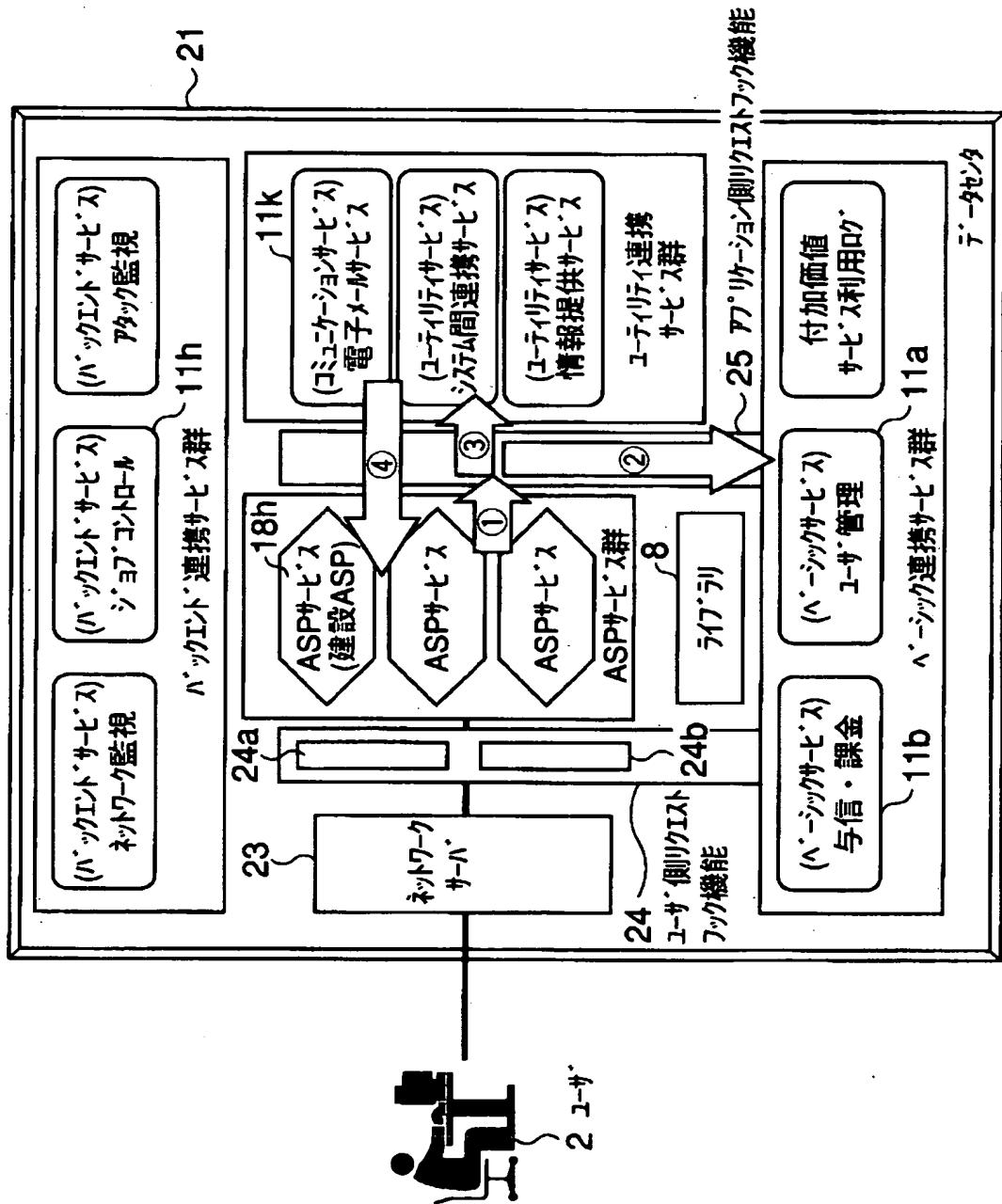
〔図4〕



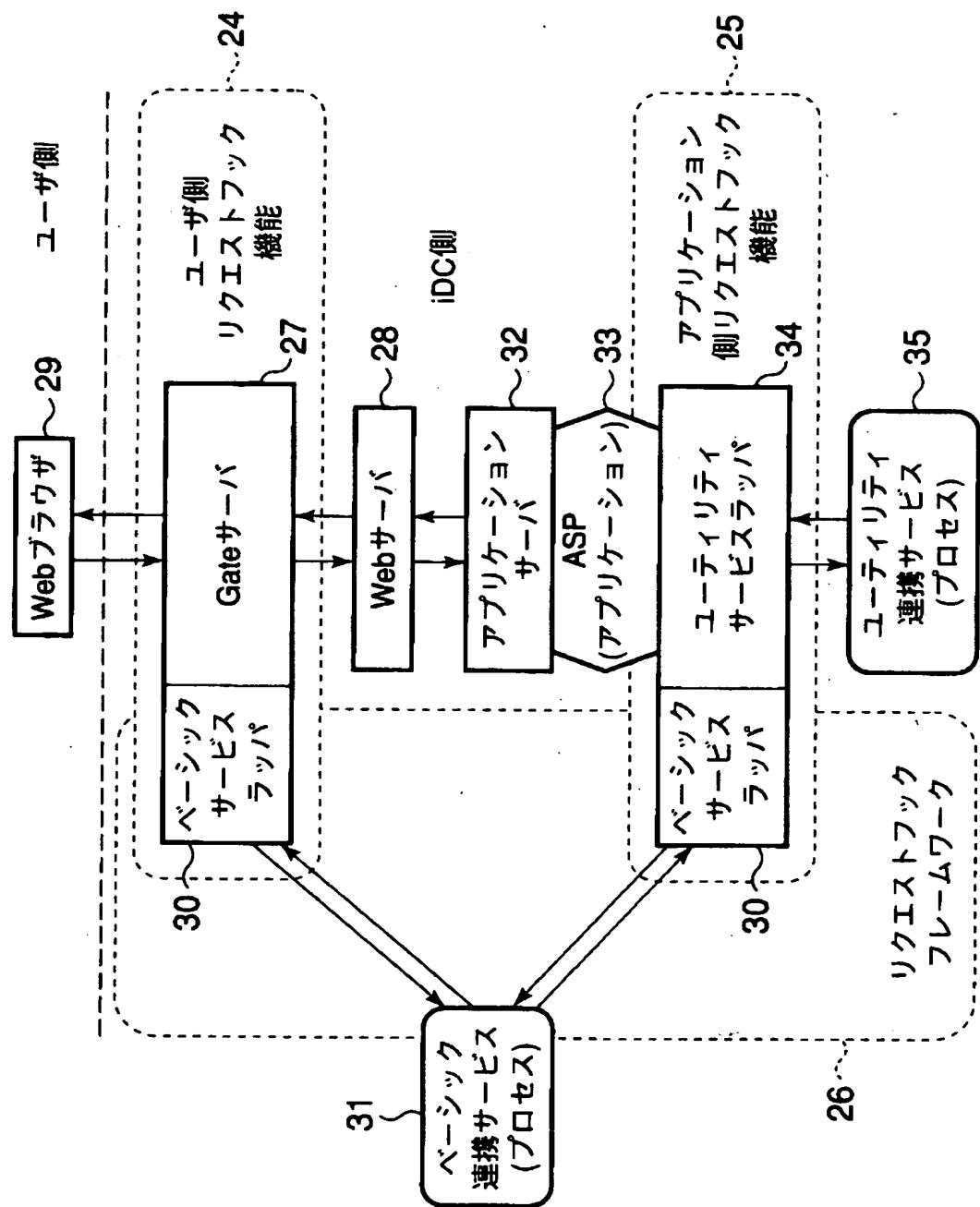
【図5】



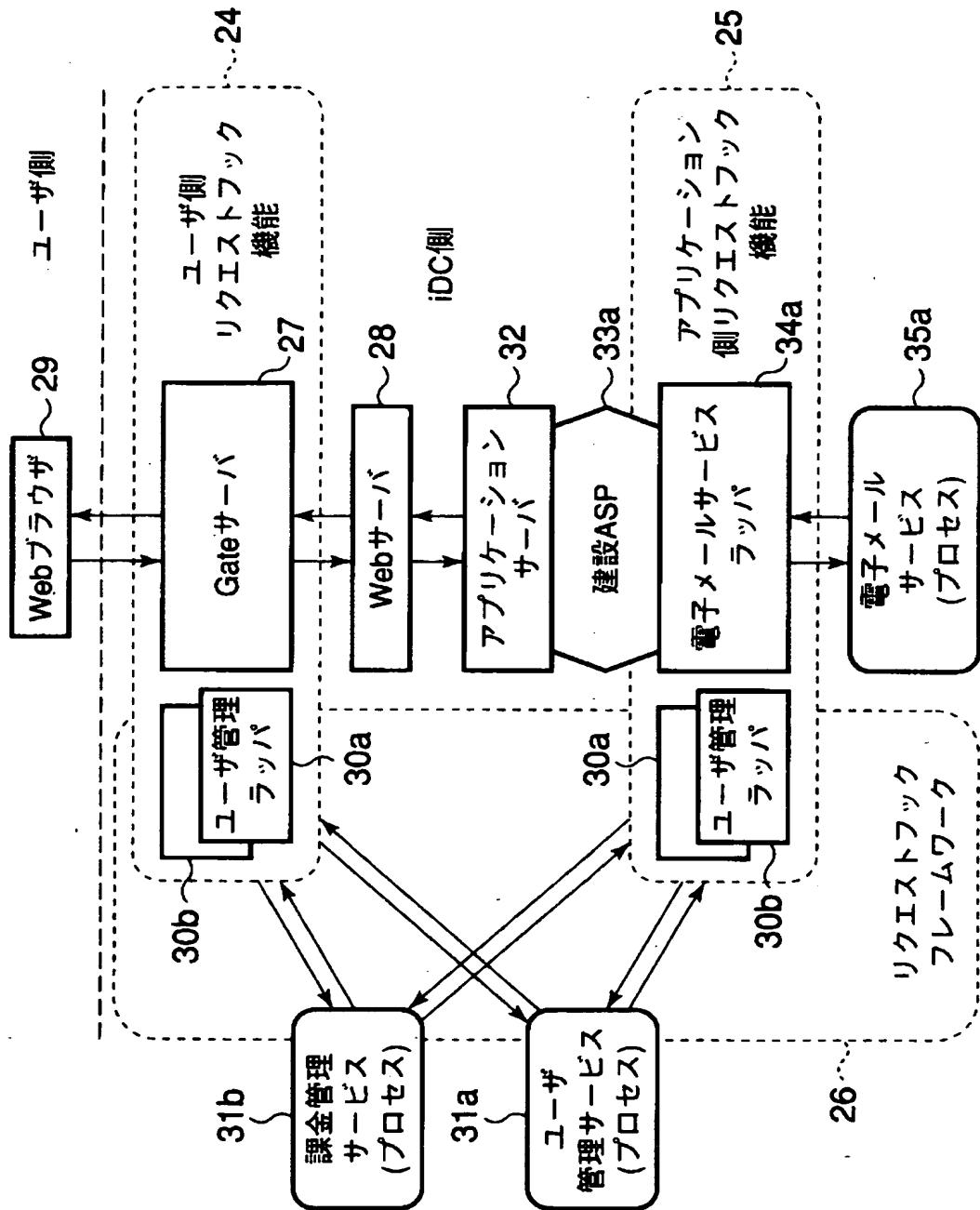
【圖 6】



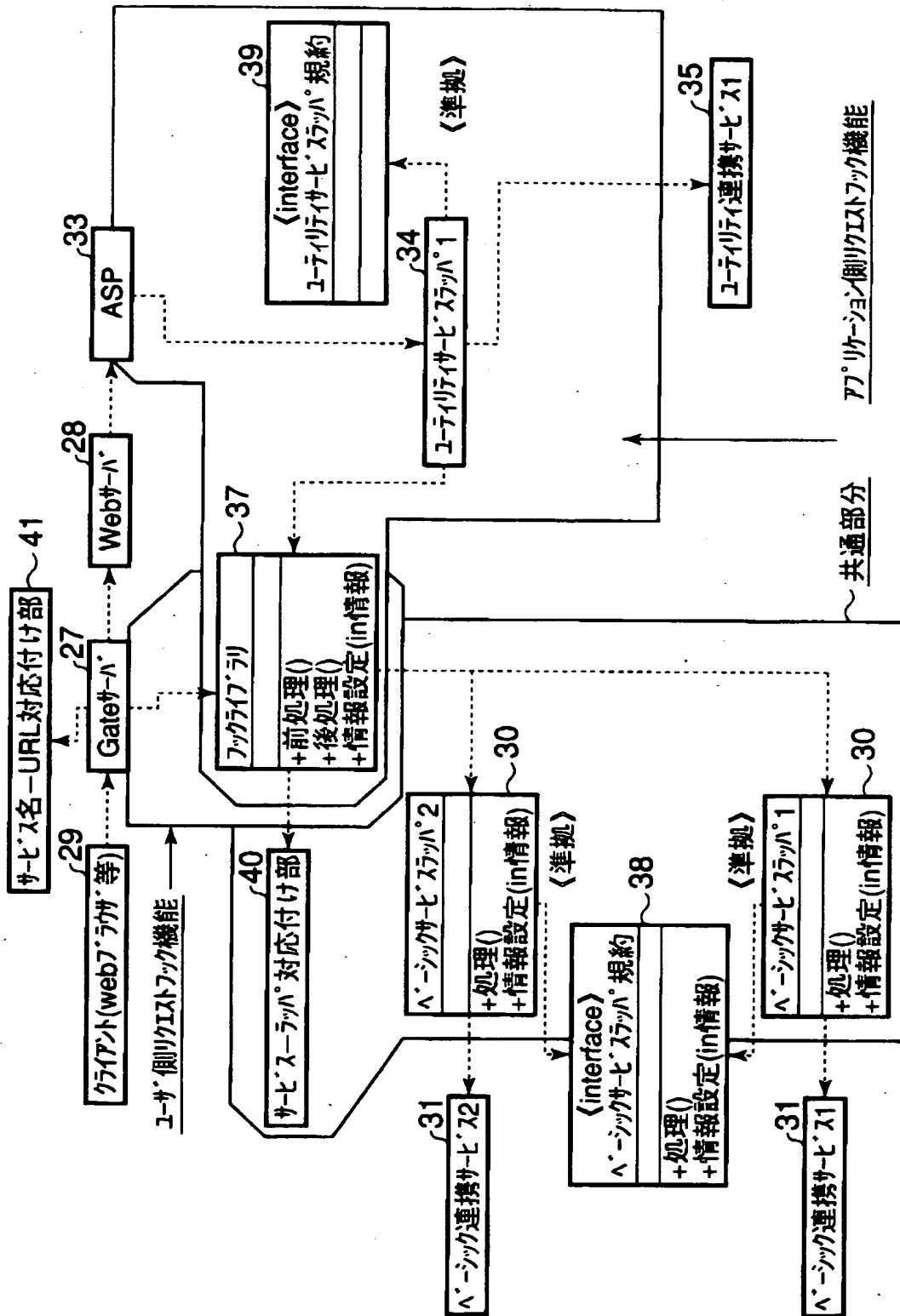
【図7】



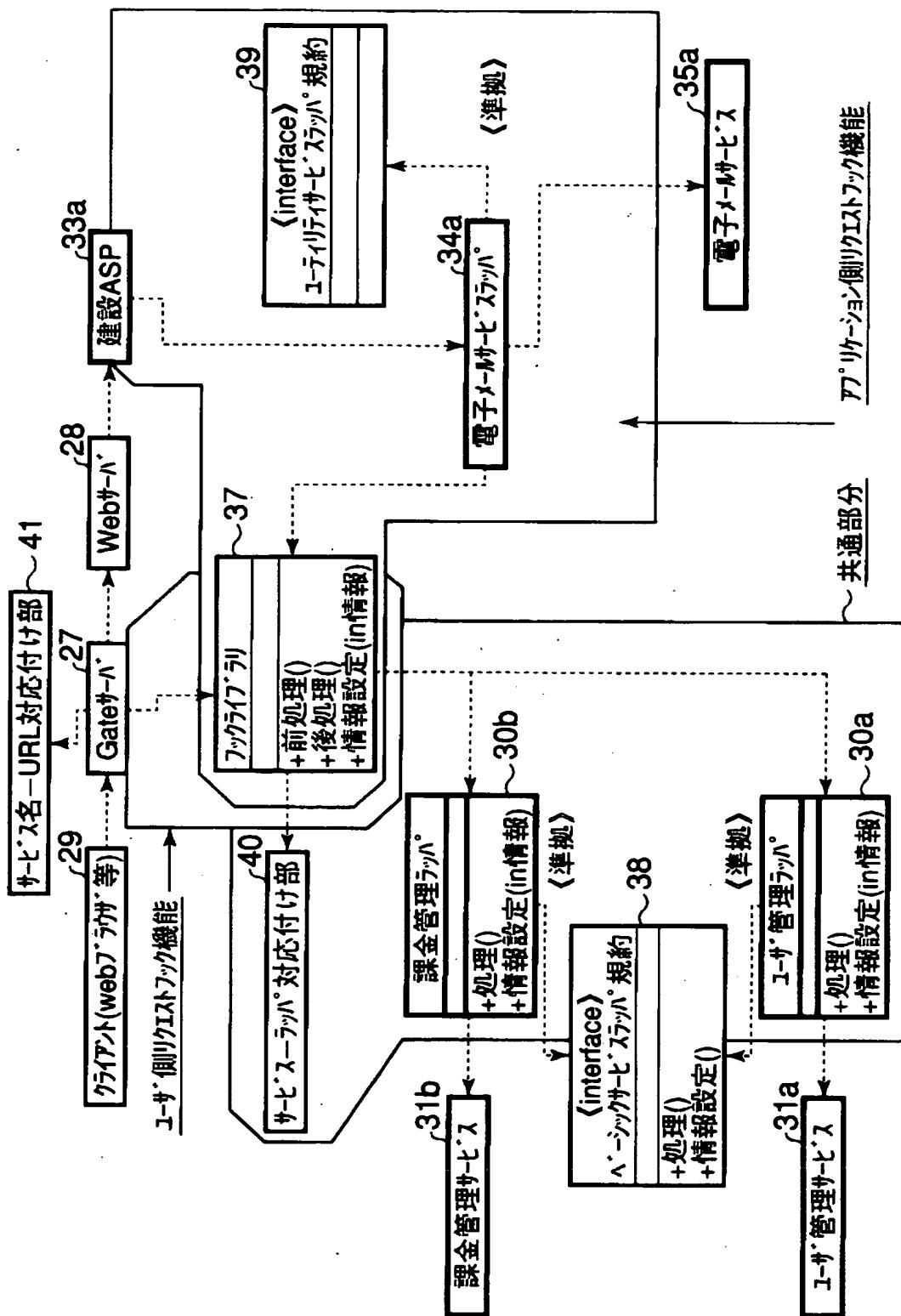
【図8】



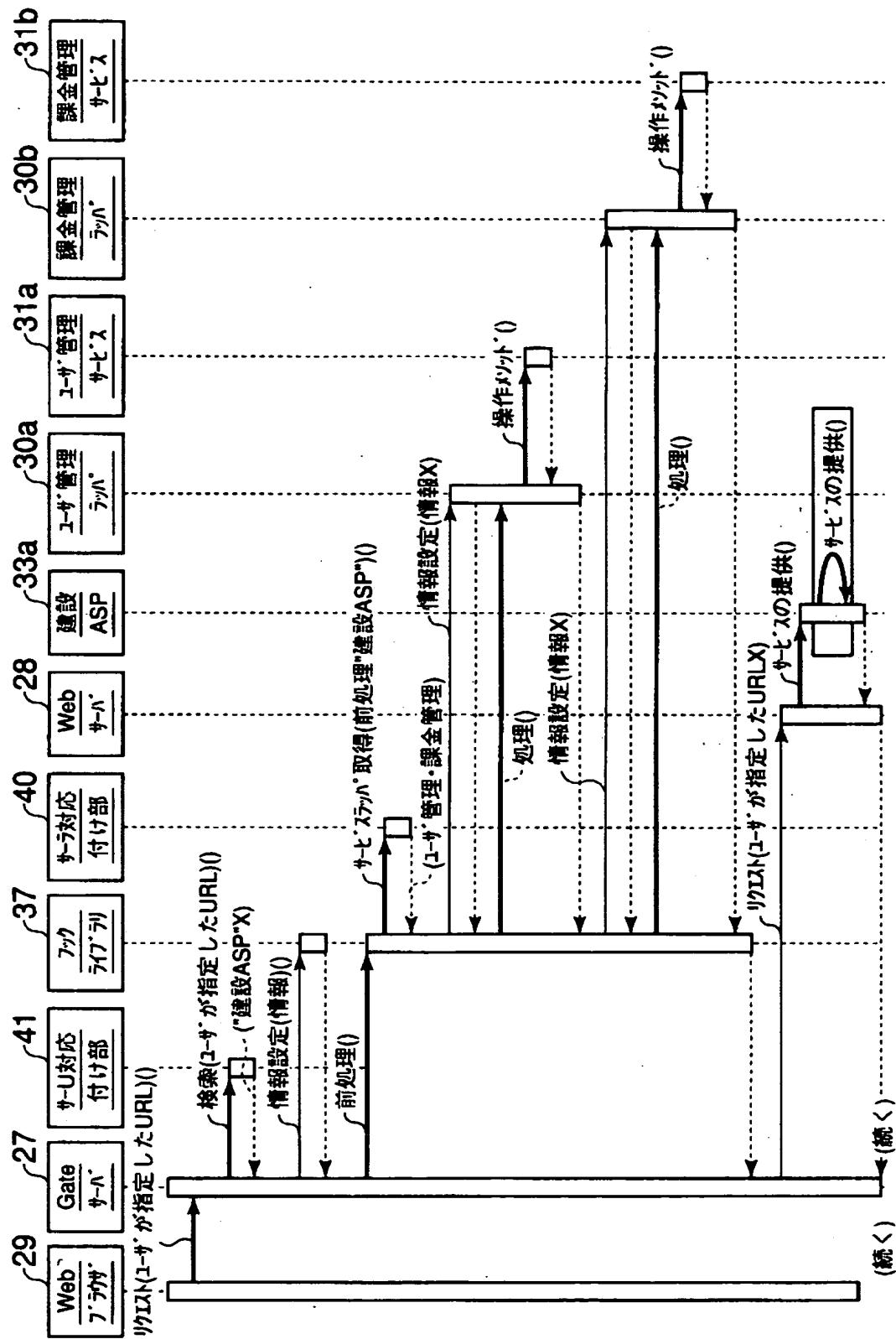
【図9】



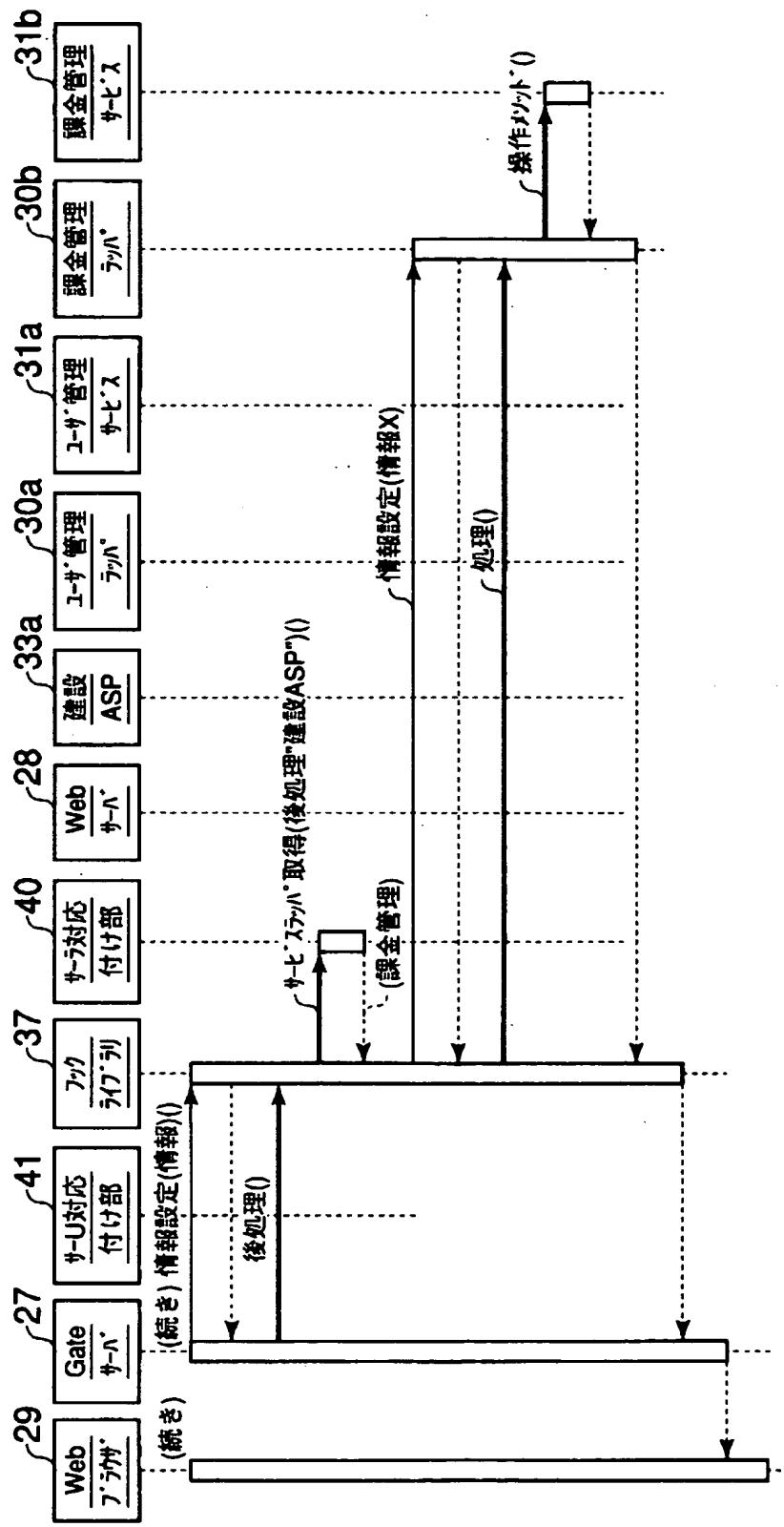
【図10】



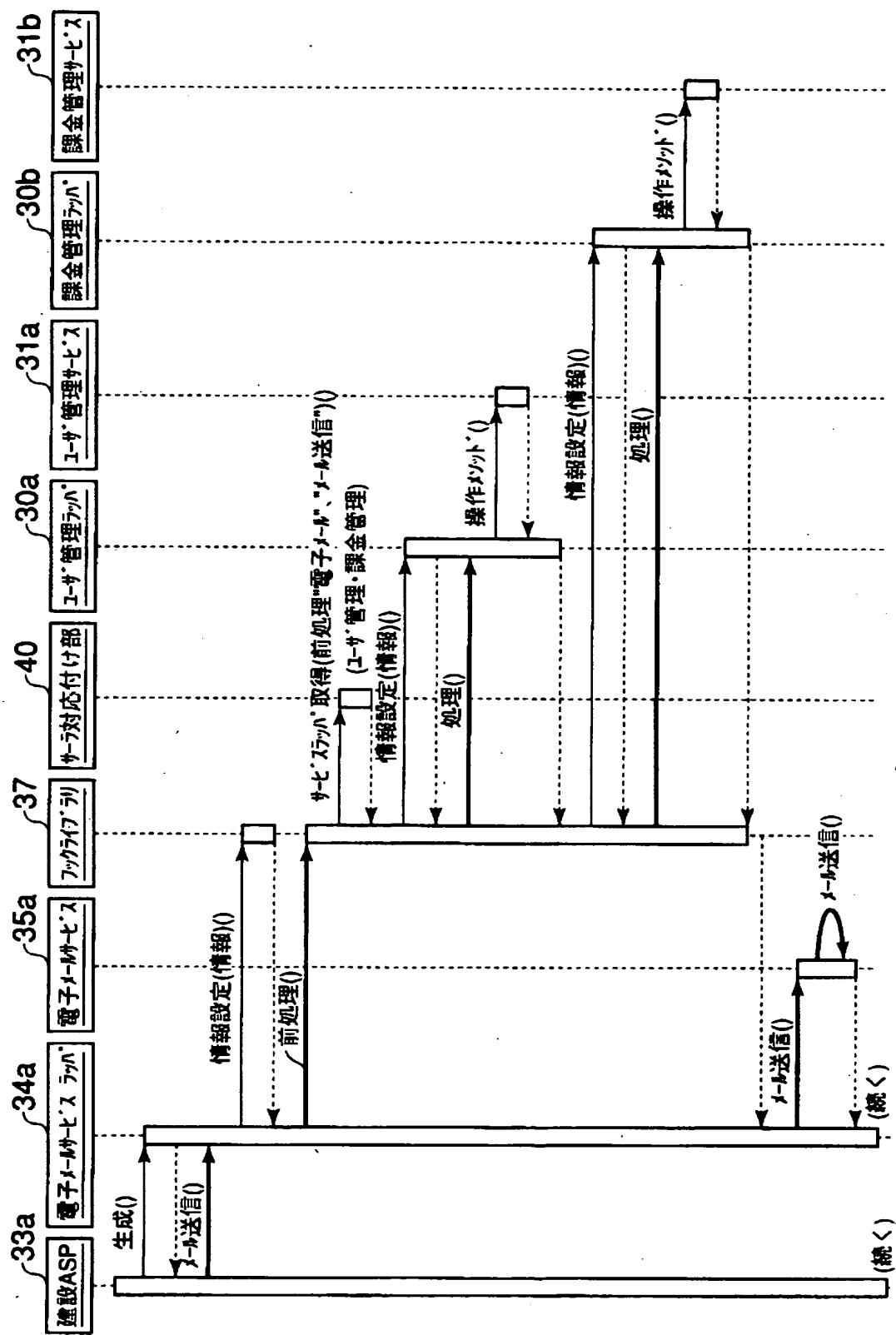
【図11】



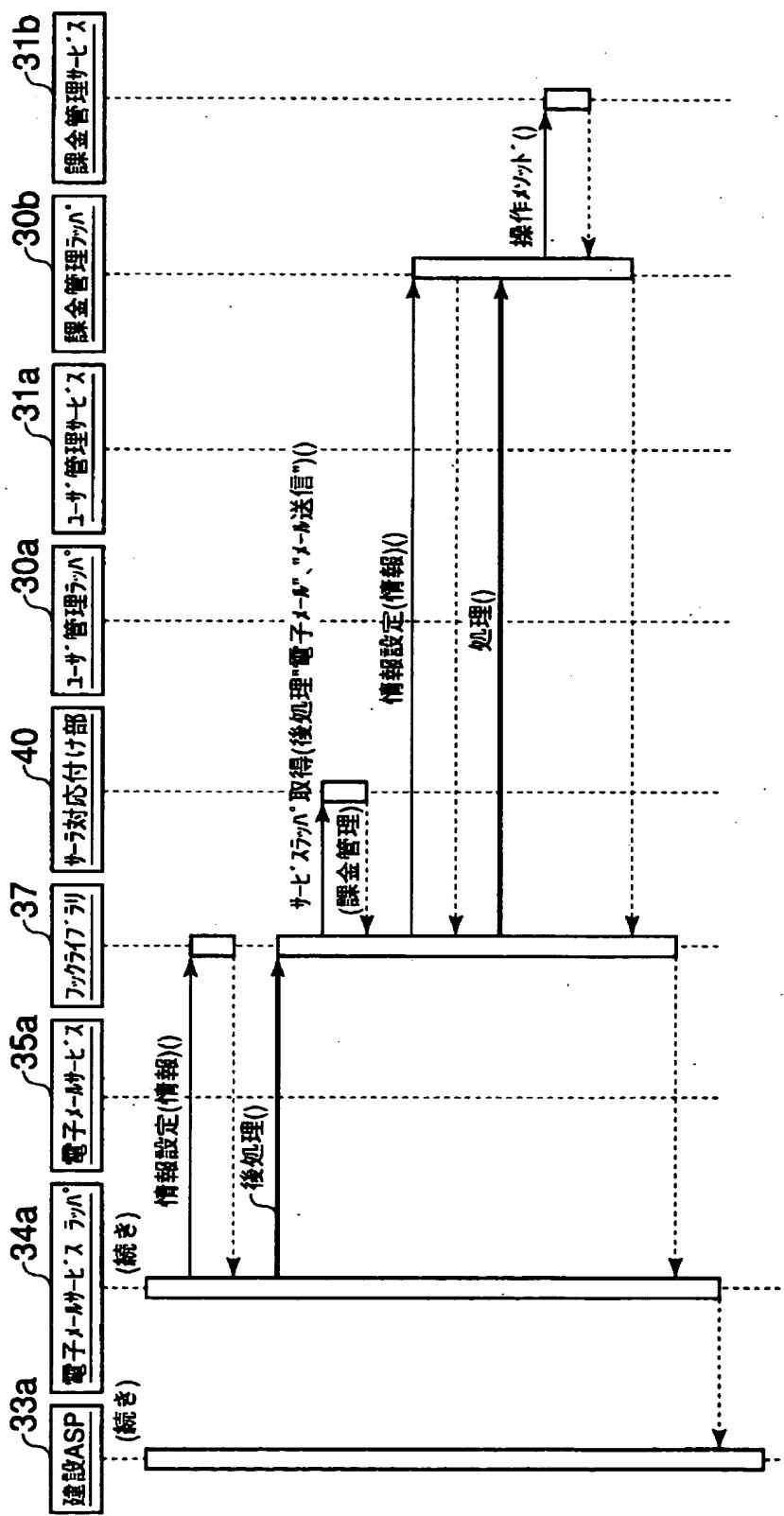
【図12】



【図13】



【図14】



【書類名】 要約書

【要約】

【課題】 ユーザに提供されるサービスを容易に実施可能とする。

【解決手段】 コンピュータシステムによる情報処理サービスの提供方法に関する発明である。本発明の情報処理サービスの提供方法では、まず、ユーザにネットワークを経由して提供されるサービスに対して付加価値サービスを提供するためのイベントを検出する。次に、イベントが検出された場合に、付加価値サービスを提供するデータ処理手段を呼び出すために用意されたライブラリを参照する。そして、ライブラリの参照結果を利用して付加価値サービスを提供するデータ処理手段に処理実行を依頼する。

【選択図】 図1

出願人履歴情報

識別番号 [000003078]

1. 変更年月日 1990年 8月 22日

[変更理由] 新規登録

住 所 神奈川県川崎市幸区堀川町72番地
氏 名 株式会社東芝

2. 変更年月日 2001年 7月 2日

[変更理由] 住所変更

住 所 東京都港区芝浦一丁目1番1号
氏 名 株式会社東芝